HETED-HÉT

commodore 64

1·2·3

ELŐSZÓ A MÁSODIK KIADÁSHOZ

Mindannyian, akik részt vettünk a Hetedhét Commodore 64 című könyv létrehozásában, nagy örömmel vettük tudomásul, hogy munkánkat a közönség is, a kritika is kedvező fogadtatásban részesítette. Olyannyira, hogy hamar elő kellett rukkolnunk a második kiadással, mert az előző alig pár hét alatt mind egy szálig elfogyott.

Olvasóink kérésére - és a nyomdai átfutási idő csökkentése érdekében - az eddigi három füzetet egy kötetben foglaltuk össze, s ahol észrevettük, javítottuk az előző kiadás hibáit is. Az új kiadás árát így alacsonyabban szabhattuk meg, mint az előzőé volt. Reméljük, olvasóink emiatt nem vonják meg rokonszenvüket a könyvtől...

Budapest, 1985. szeptember

Novotrade RT.

Pál Zsuzsanna - Révbíró Tamás

HETEDHÉT
Commodore 64

ELSÖ HÉT



NOVOTRADE RT. 1985

Programok: Pál Zsuzsanna
összekötő szöveg: Révbíró Tamás
Illusztrációk, műszaki szerkesztő: Dévényi Erika

Szakmai lektor: Baumann Gábor
Szerkesztő: Marosváry Tamás

Felelős kiadó: Rényi Gábor igazgató

Ez a könyv a
Precision Software Ltd.
Easy Script szövegszerkesztő programja
segítségével készült
Commodore 64 számítógépen,
Commodore 1526-os printeren
A programlisták
Commodore MPS 801-es printeren készültek

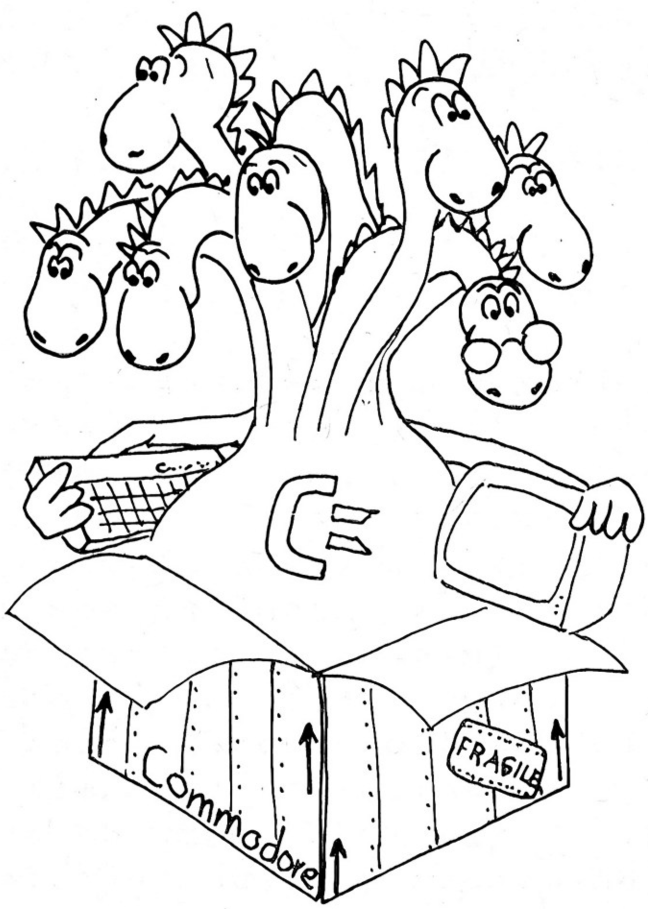
ISBN 96302 3637 7
HU ISSN 0237-3041

Készült a Somogy Megyei Nyomdaipari Vállalat
Kaposvári üzemében - 85-6687
Felelős vezető Mike Ferenc igazgató

ELŐSZÓ

Előszó tulajdonképpen nem is kellene, de hát annyi könyvben van, hogy arra gondoltunk: mi is írunk egyet. Pedig tulajdonképpen ez nem is könyv olyan nagyon - hiszen a könyvek többnyire csupa komoly és okos dologról szólnak, a miénk meg arról, hogy hogyan lehet egy nagyon komoly és okos dologgal, a számítógéppel - játszani.

A szerzők



Az úgynevezett személyi (más néven: mikro) számítógépek közül a Commodore 64 az egyik legelterjedtebb és legtehetségesebb. Sok mindent tud - még elromlani is, de egyet mindjárt az elején meg kell mondani: a gép attól sohasem romlik el, ha csak a billentyűket nyomkodja az ember; mindegy, hogy milyen billentyűt nyomunk meg rajta, milyen sorrendben. Legföljebb nem érti, mit akarunk tőle - és ezt többnyire közli is...

Amikor a gépet rendesen összekötöttük a monitorral (ez lehet egy akármilyen tévékészülék is, de azért jobb, ha színes, és még jobb, ha a géphez tartozó monitor), és bekapcsoljuk, a képernyőn világoskék keretben, sötétkék alapon, világoskék betűkkel a következő feliratot látjuk:

```
    **** COMMODORE 64 BASIC V2 ***

 64K RAM SYSTEM 38911 BASIC BYTES FREE

READY.
█
```

Ennek nagy részével egyelőre nem sokat kell törődni - tekintsük címlapnak, amely arról tájékoztat, hogy milyen számítástechnikai rendszer áll rendelkezésünkre. Egy apró dologgal kell csupán foglalkoznunk: azzal a világos kis négyzettel, amelyik a keret mellett a bal szélen villog; ennek még nagy szerepe lesz.

A villogó jel neve angolul cursor (van, aki latinosan kurzornak mondja, van, aki angolosan korszornek - ez csupán szokás dolga), és az a szerepe, hogy mindig pontosan tájékoztasson, hol járunk a képernyőn, vagyis ha megnyomunk egy billentyűt, akkor az annak megfelelő jel hol fog megjelenni. Olyan ez az

egész, mint egy írógép, amelyik papír helyett a képernyőre ír.

Próbáljuk is ki. Nyomjuk meg a számokkal vagy betűkkel ellátott billentyűk közül akármelyiket - látni fogjuk, hogy a cursor egy hellyel jobbra ugrik, és helyén az a jel áll, amelynek a billentyűjét megnyomtuk.

Most bal kézzel nyomjuk meg a SHIFT feliratú gombot (baloldalt alul a második), és anélkül, hogy ezt a gombot elengednénk, nyomjuk meg a CLR/HOME feliratú gombot is.

Mi történt? A képernyőről minden felirat, szöveg és jel eltűnt, csak a cursor maradt meg, de az is fent van a képernyő bal felső sarkában (ezt nevezzük alaphelyzetnek).

Jegyezzük meg: ha a SHIFT és a CLR/HOME gombot együtt megnyomjuk, a képernyő üres lesz, és a cursor alaphelyzetbe kerül.

Azt mondtuk, úgy működik a gép, mint egy írógép. Bizonyságul írjuk fel a nevünket a képernyőre! Sorban nyomjuk meg a megfelelő betűket! (Akinek ékezetes betű van a nevében, az most felejtse el az ékezetet - a gépen csak az angol ábécé betűi találhatók meg. Ezt azonban eléggé könnyű megszokni; csak egy dolog fontos: hogy emiatt senki se felejtse el a magyar helyesírást!)

Ha elkészültünk, akkor a képernyőn ilyesforma felirat lesz látható:

```
*********************
*KOVACSMARCI█       *
*                   *
*                   *
```

Hoppá! Nem hagytunk szóközt a két név között. Semmi baj, a számítógépen radír nélkül is lehet javítani. Csak az kell hozzá, hogy a cursort arra a helyre vigyük, ahol javítani akarunk a szövegben.

A billentyűzeten jobboldalt alul van két gomb, mindkettőn CRSR felirat áll és két-két nyíl. Az egyikkel le és fel, a másikkal balra és jobbra lehet mozgatni a cursort, anélkül, hogy a képernyőre írt szöveggel bármi történne. A bal oldali CRSR gomb önmagában lefelé viszi a cursort, ha a SHIFT billentyűt is megnyomjuk vele együtt, akkor pedig fölfelé. A másik

CRSR billentyű jobbra viszi a cursort, SHIFTtel együtt pedig balra. Gyakoroljuk egy kicsit, kószáljunk a képernyőn ide-oda, aztán menjünk vissza a nevünkhöz, ott is a keresztnév utolsó betűje utáni első helyre. Most nyomjuk meg a jobb felső sarokban lévő, INST/DEL feliratú billentyűt. A szövegből ennyi marad: KOVACSMARC■. Ahányszor megnyomjuk ezt a billenyűt, annyiszor lép egyet balra a cursor, és letörli a képernyőről, ami az útjába kerül.

Ha letöröltük az egybeírt keresztnevet, lépjünk egy helyet jobbra a cursorral, és beírhatjuk újból, ahogy az előbb.

Most tehát ilyen a képernyőnk:

```
**********************
*KOVACS MARCI■       *
*                    *
*                    *
```

Próbáljunk ki még valamit! Menjünk vissza a cursorral a keresztnév első betűjére, nyomjuk meg a SHIFT gombot, és vele együtt az INST/DEL feliratú billentyűt. Ekkor az egész MARCI név egy hellyel jobbra ugrik, és nagyobb lesz a hézag a két név között. Jegyezzük meg: ha egy-két kimaradt betűt, jelet akarunk a szövegbe betoldani, így tudunk helyet csinálni neki.

Lépkedjünk most a cursorral a vezetéknév utáni szóközre, majd meg egy hellyel tovább, és írjunk be egy másik keresztnevet! Legyen ez BARNABÁS. Ne ijedjunk meg attól, hogy a sor nem üres: az újonnan beírt betűk kitörlik az előzőeket; mindig az érvényes, amit utolsónak írtunk be.

Most ilyen a képernyő:

```
**********************
*KOVACS BARNABAS■    *
*                    *
*                    *
```

Most menjünk vissza újból a keresztnév első betűjére, és javítsuk ki erre: PÁL. Ha beírjuk a hárombetűs nevet, ugyancsak furcsa képet látunk:

```
**********************
*KOVACS PAL█ABAS     *
*                    *
*                    *
```

Semmi baj - nyomjuk meg néhányszor a legalsó, hosszú, vízszintes billentyűt (ez a szóközbillentyű, ugyanúgy, mint a közönséges írógépeken), és a fölösleges betűk sorban eltűnnek. Ez a billentyű eggyel jobbra ugratja a cursort, és közben kitörli, ami az útjába akad. (Használata ezért nagyobb óvatosságot is követel, mint a CRSR gomboké.)

Most, hogy a cursort ilyen nagyszerűen, magabiztosan tudjuk kezelni, elszórakozhatunk egy kicsit azzal, hogy a képernyőn mindenfelé kiírogatjuk a nevünket. (Eközben megfigyelhetjük, hogy ha elérjük a képernyő jobb szélét, akkor a cursor automatikusan a következő sor elejére ugrik - mennyivel kényelmesebb ez, mint az írógép!)

Most egy új játék következik. Nyomjuk meg a billentyűzet bal alsó sarkában lévő gombot (amelyen a C= jel, a Commodore gépek emblémája látható), és ugyanakkor nyomjuk meg a SHIFTet is!

A képernyőn minden betű kisbetűvé változott! Jó tudni, hogy a számítógépen ilyen lehetőség is van (bár általában a nagybetűket használjuk), ez azonban még önmagában nem játék. A játék most jön.

Tartsuk benyomva a SHIFT billentyűt, vagy - ami ugyanolyan hatású - nyomjuk meg a SHIFT/LOCK feliratú billentyűt. Igy írjunk valamilyen szöveget a képernyőre. Ha ezután megnyomjuk a C= gombot, a szöveg átváltozik mindenféle fantasztikus krikszkrakszszá. Valóságos titkosírás ez - és megfejteni ugyanolyan könnyű: ismét be kell nyomni a C= gombot.

Ennek magyarázata a következő: a számítógép alapállapotban - amikor először bekapcsoljuk - a billentyűk tetején látható betűket, jeleket írja a képernyőre. Ha benyomjuk a SHIFT billentyűt, akkor a billentyűk oldalán jobboldalt látható jelek kerülnek a képernyőre, ha pedig a C= (Commodore) gombot nyomjuk meg, akkor a bal oldaliak.

Ott, ahol a billentyű tetején két jel van egymás alatt (például a % jel és az 5 számjegy), ott a felsőt a SHIFT billentyű segítségével tudjuk előcsalni. (Azt pedig, hogy a SHIFT és a C= billentyű együttes megnyomására a nagybetűk kisbetűvé változnak, az előbb már láttuk.)

ÖSSZEFOGLALÓ KÉRDÉSEK, FELADATOK

1. Próbáljunk meg a billentyűk oldalán látható jelekből rajzolni valamit a képernyőn - például egy házikót!

2. Játsszunk a képernyőn amőba játékot! (Alig hisszük, hogy van, aki ne ismerné; ha mégis akad ilyen, az akárkitől megkérdezheti, mik a szabályai.) Az egyik játékos az X betűt használhatja a játékban, a másik az O betűt vagy a Q billentyű oldalán látható (SHIFTtel előhívható) pontot.

3. Játsszunk betűpiramist! Takarítsuk ki a képernyőt (SHIFT és CLR/HOME), aztán írjunk be egy kétbetűs szót, szorosan a bal felső sarokba. A következő lépésben egy hárombetűs szót kell beírni, de úgy, hogy a már beírt két betű sorrendje ne változzon meg - tehát vagy a két betű elé, vagy közé, vagy mögé írhatunk.

A hárombetűs után négybetűs következzék, ugyanilyen feltételek mellett, majd ötbetűs - és így tovább. Ez nyelvi játéknak sem utolsó - de közben gyakoroljuk a cursor meg a szöveg mozgatását is...

Egy példa:

HA
HAT
IHAT
KIHAT
KIHAJT
KIHAJIT
KIHAJLIT



Amikor bekapcsoljuk a gépet, kék háttéren kék betűk tűnnek fel, kék keretben. Nem érne azonban sokat az egész, ha kötelező volna mindig ezt a kékséget bámulni; lehet, hogy nem is tesz nagyon jót az ember szemének. Szerencsére nagyon könnyen javíthatunk rajta. Takarítsuk ki a képernyőt (SHIFT és CLR/HOME), majd a bal felső sarkába gépeljük be a következőket:

```
POKE 53280,8
```

(Aki tud írógépelni, az itt nagyon vigyázzon: a számítógépen külön van 1-es és 0, tehát egyes helyett véletlenul se gépeljünk kis l betűt, 0 helyett O-t!)

Most pedig nyomjunk meg egy olyan gombot, amelyikhez eddig még nem nyúltunk: a harmadik billentyűsor jobb szélén található, RETURN feliratút!

A képernyőn szembetűnő vátozás történt: a keret színe barnává változott!

Töröljük le ismét csak a képernyőt (SHIFT+CLR/HOME), és gépeljük be a következő szöveget:

```
POKE 53281,8
```

Most valami apróbb baj történt: a képernyő színe tetszetős narancssárga lett ugyan, de a betűk abszolúte nem olvashatók - csak valami világoskék maszat van a helyükön.

Ezen sem nehéz segíteni: nyomjuk le a második billentyűsor bal szélén található, CTRL feliratú billentyűt, és anélkül, hogy elengednénk, nyomjuk meg az 1-es számjegy billentyűjét (a legfelső sorban balról a második)!

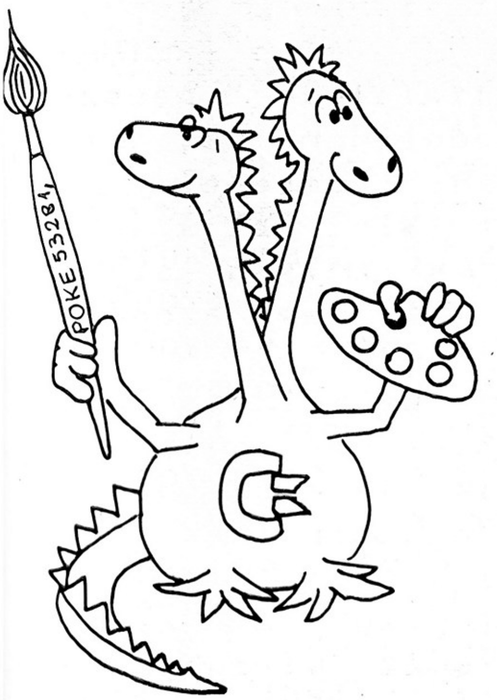
A cursor színe fekete lett - a narancssárga alapon remekül látható. Mostantól, ha tetszik, ebben a

színösszeállításban használhatjuk a képernyőt. Ha nem tetszik, mindenki magának beállíthatja a képernyő színeit olyanra, ahogyan számára a legkellemesebb.

Előfordulhat, hogy a színválogatás közben egyszer csak azt vesszük észre, hogy a képernyő teljesen üres lett - még a cursort sem találjuk meg rajta. Erre az a magyarázat, hogy véletlenül ugyanolyan színűre állítottuk a cursort, mint a képernyő színe. Ilyenkor is megvan tehát, csak nem látszik. Fehér alapon fehér cursorral is lehet dolgozni; ha nem követünk el gépelési hibát, a gép ilyenkor is megérti, mit akarunk tőle, csak éppen semmi értelme a dolognak, hiszen önmagunkat nem tudjuk ellenőrizni. Röviden tehát: ha egyszer csak eltűnik a cursor, próbáljuk meg más színűre állítani!

(A cursor színének megváltoztatásához nemcsak a CTRL billentyűt, hanem a C= jelűt is használhatjuk: ha például a C= gombot és a 2-es számjegy billentyűjét egyszerre lenyomjuk, a cursor színe barna lesz - ugyanolyan, mint a keret színe.)

Ehhez a következő táblázat nyújt segítséget.



SZÍNKÓD

0	FEKETE	8	NARANCS
1	FEHÉR	9	BARNA
2	PIROS	10	RÓZSASZÍN
3	CIÁNKÉK	11	SÖTÉTSZÜRKE
4	BÍBOR	12	KÖZÉPSZÜRKE
5	ZÖLD	13	VILÁGOSZÖLD
6	KÉK	14	VILÁGOSKÉK
7	SÁRGA	15	VILÁGOSSZÜRKE

A KERETSZÍN MEGHATÁROZÁSA
POKE 53280,SZÍNKÓD

A HÁTTÉRSZÍN MEGHATÁROZÁSA
POKE 53281,SZÍNKÓD

A CURSOR SZÍNÉNEK MEGHATÁROZÁSA

CTRL SEGÉDBILLENTYŰVEL	C= SEGÉDBILLENTYŰVEL
1 FEKETE	1 NARANCSSÁRGA
2 FEHÉR	2 BARNA
3 PIROS	3 RÓZSASZÍN
4 CIÁNKÉK	4 SÖTÉTSZÜRKE
5 BÍBOR	5 KÖZÉPSZÜRKE
6 ZÖLD	6 VILÁGOSZÖLD
7 KÉK	7 VILÁGOSKÉK
8 SÁRGA	8 VILÁGOSSZÜRKE

Ezt a táblázatot a könyv végén is megtalálhatjuk; mire odáig elérünk az olvasásban (meg a gyakorlásban), már sokkal áttekinthetőbb lesz. Most csak azért írtuk ide, hogy a következő játékhoz ne kelljen sokat lapozgatni.

A játék lényege: írjuk fel a képernyő bal felső sarkába a képernyő színének megváltoztatására szolgáló jelszót: POKE 53281, (figyeljünk a vesszőre!) - aztán írjunk be egy tetszés szerinti számot 0 és 15 között! Utána egy darabig ne nyomjuk le a RETURN billentyűt, hanem próbáljuk meg kitalálni (persze puskázás nélkül), hogy milyen színt kódoltunk be. Ezt ketten is játszhatják - az egyik begépeli a színkódszámot, a másik megpróbálja kitalálni a színt. Ha megmondta, hogy szerinte a beírt szám milyen színt jelent, megnyomjuk a RETURN gombot, és a gép azonnal igazolja, hogy helyes volt-e a tipp.

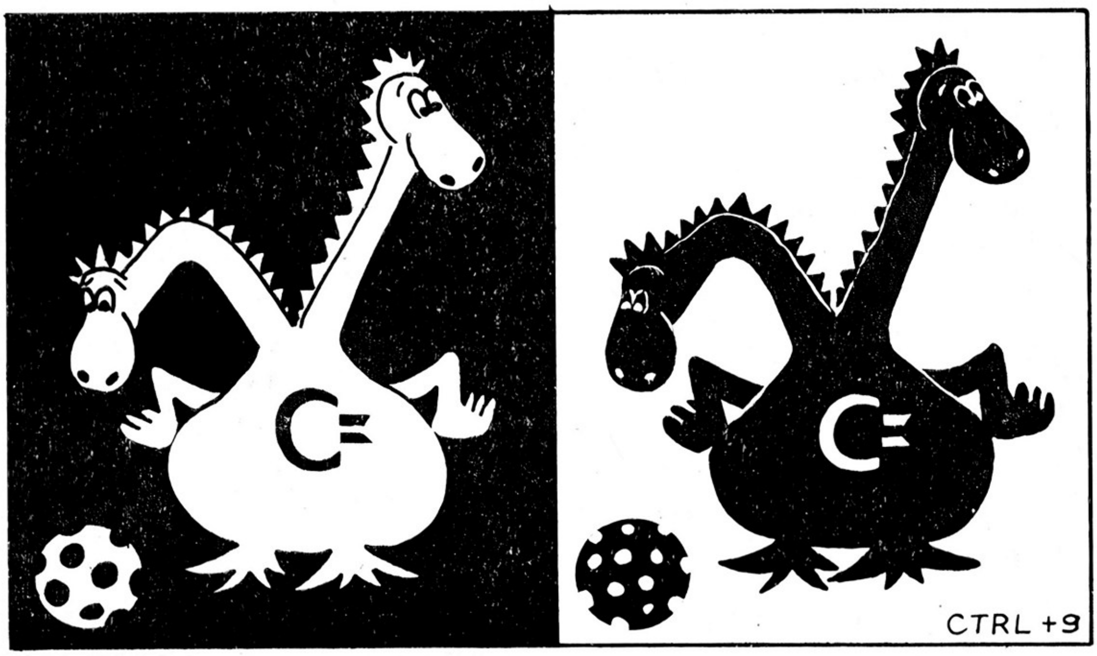
Egy másik játék: az egyik játékos felírja egy cédulára, hogy milyen színű legyen a képernyő, milyen a keret, milyen színű a cursor, a másik játékos pedig begépeli a szükséges adatokat, aztán cserélnek. Az győz, aki rövidebb idő alatt tudja teljesíteni a partner kívánságát. (Egy példa: legyen a keret világoskék, a képernyő kék, a cursor ismét világoskék! Ehhez a következőket kell beírni:

```
POKE 53280,14
POKE 53281,6
```

után pedig megnyomni a C= jelű gombot, és vele együtt a 7 számjegy billentyűjét. És íme, máris előállt az a képernyőszín-összeállítás, amelyik a gép bekapcsolásánál magától "bejelentkezik"...)

Újabb játék következik. Most, ha az utasításokat pontosan követtük, a képernyő színe sötétkék, a keret és a cursor világoskék.

Takarítsuk ki a képernyőt (SHIFT+CLR/HOME)! Nyomjuk meg a CTRL feliratú gombot, és vele egyszerre a 9.számjegy billentyűjét!



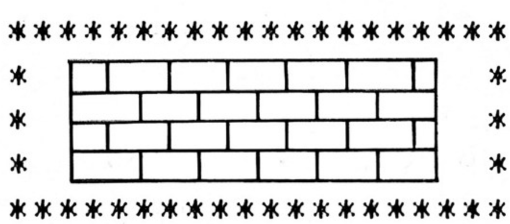
Látszólag nem történt semmi. De próbáljunk meg most begépelni valamilyen szöveget! Látni fogjuk, hogy a betűk nem úgy jelennek meg, ahogyan megszoktuk: a képernyő szinénél világosabb színben, hanem világos alapon, sötéten – más szóval negatívban. (Aki foglalkozott már fotózással, az tudja, mit jelent a "negativ" szó: azt, hogy ami a valóságban világos, az sötét lesz és megfordítva.) Ezek a negatív jelek (angolul inverse vagy reverse – a 9-es számjegy és a 0 billentyűjén ez utóbbi szó rövidítése látható) számtalan alkalommal lesznek segítségünkre.

Inverz módban vagyunk tehát, egy idő óta a cursor színét is tetszés szerint tudjuk változtatni – semmi akadálya tehát, hogy szép szivárványszínű csíkokat húzzunk a képernyőn. Ehhez nem kell más, mint megnyomni és nyomva tartani a szóközbillentyűt. Ha a cursor világoskék volt, akkor a csík is világoskék

lesz, de ha bárhol megváltoztatjuk a színét, akkor attól a helytől kezdve az új színnel folytatódik. Ha pedig megúntuk a negativ karaktereket, a CTRL és a 0 billentyű együttes megnyomásával visszaléphetünk a megszokott módba.

Nagyon csalódnánk, ha erre nem lehetne valami játékot kitalálni. Az egyik legegyszerűbb: rajzolja meg ki-ki a képernyőn a magyar zászlót! Inverz módba lépünk: a CTRL és a 9-es billentyűt nyomjuk meg egyszerre; utána pirosra változtatjuk a cursort: CTRL és a 3-as billentyű; aztán a szóközbillentyűt addig tartjuk nyomva, ameddig a piros sáv el nem foglalja a képernyő egyharmadát. Ekkor - természetesen egy sor elején - fehérre változtatjuk a cursort: CTRL és a 2-es billentyű; és a képernyő középső harmadát átszínezzük fehérre. Az utolsó harmad elején zöldre változtatjuk a cursort: CTRL és a 6-os billentyű, és a képernyő utolsó sorait zöldre színezzük. Vigyázzunk arra, hogy az utolsó előtti helyen álljunk meg a cursorral, mert ha a képernyő utolsó betűhelyét is betöltjük, akkor a képernyő tartalma két sornyit megemelkedik, és felül a fáradságosan kifestett piros sorainkból kettő visszahozhatatlanul elvész.

A magyar zászló után természetesen megpróbálkozhatunk más országok zászlóival is; a negativ és pozitiv jelek ügyes váltogatásával, keverésével egészen bonyolult grafikákat is készíthetünk - akár a libanoni zászlón szereplő cédrust is lerajzolhatjuk. (Emlékezzünk vissza az előző fejezetben tárgyalt grafikus jelekre, amelyek a SHIFT és a C= billentyű segítségével csalhatók elő - ezeket bátran használhatjuk mind pozitiv, mind negatív változatban!)
Egy másik játék: rajzoljunk egy téglafalat a képernyőre piros téglákból, fehér habarccsal.



Milyenre kell változtatni a képernyő színét, milyenre a cursorét? Magyarán: piros alapon fehérrel vagy fehér alapon pirossal érdemes elkezdeni a rajzolást?

ÖSSZEFOGLALÓ KÉRDÉSEK

1) Mi történik, ha egyszerre megnyomjuk a

SHIFT és a C= billentyűt?

a. A képernyő színe megváltozik?

b. A nagybetűk kisbetűvé változnak és megfordítva?

c. A nagybetűk grafikus jelekké változnak?

CTRL és RVS ON billentyűt?

a. Ettől kezdve negatív jeleket ír a gép?

b. A cursor színe megváltozik?

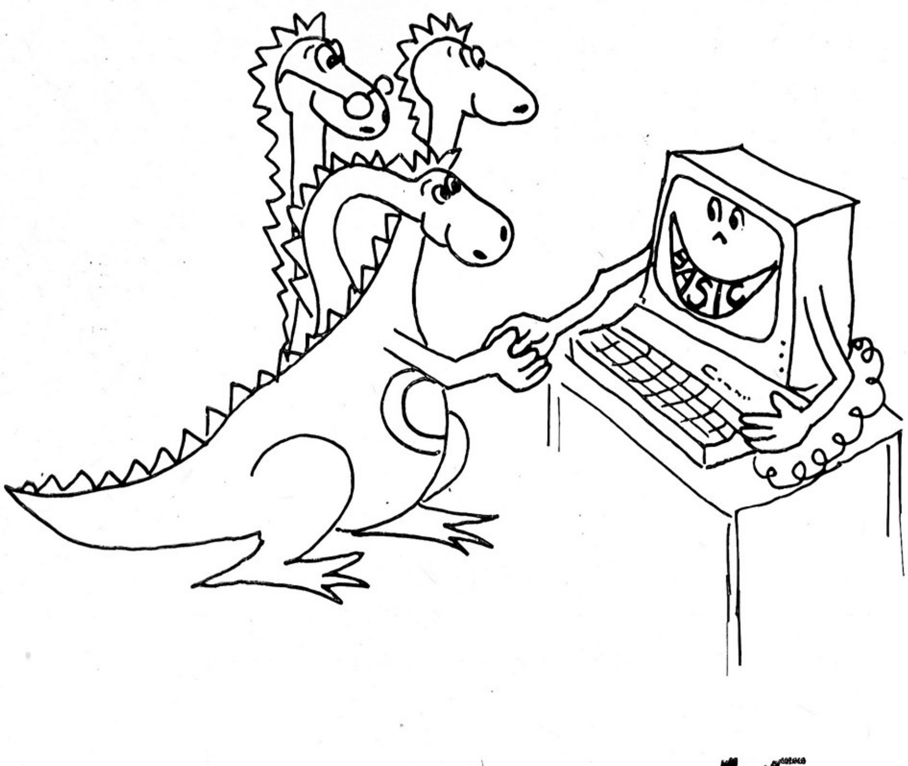
c. A képernyőről minden jel eltűnik, és a cursor alaphelyzetbe ugrik?

2) Hogyan tudunk a képernyőn "kockás" - négyzethálós - füzetlapot rajzolni, fehér alapon, fekete vonalakkal? (Rajzoljuk is meg!) Természetesen több megoldás lehetséges; ha időnk engedi, ne is érjük be eggyel!

Az előző fejezetben egyszer már megnyomtuk a RETURN billentyűt, amikor a képernyő és a keret színét változtattuk. Begépeltük a POKE 53280,satöbbit, és utána a RETURN billentyű megnyomásával utasítottuk a számítógépet a parancs végrehajtására.

Ha ezt a RETURN billentyűt nem nyomjuk le, a számítógép úgy viselkedik, mint egy különleges írógép, amely papír helyett a képernyőre ír. A RETURN gomb hatására kezd el a számítógép foglalkozni azzal, amit begépeltünk - ekkor már nem mindegy, mit írunk. Van, amit a számítógép megért, és van, amit nem.

Gépeljük be azt a szót: COMMODORE! A számítógép hűségesen kiírja a képernyőre a szót, ahogy eddig is mindent. De most nyomjuk meg a RETURN billentyűt! A számítógép azonnal válaszol. A képernyő ilyen lesz:




```
********************
*COMMODORE         *
*                  *
*?SYNTAX ERROR      *
*READY.            *
*■                 *
*                  *
```

A számítógép üzenete, a SYNTAX ERROR azt jelenti: nyelvtani hiba. A gép ugyanis csak akkor érti meg az utasításokat, parancsokat, ha egy bizonyos nyelvtan szabályai

szerint fogalmazzuk meg őket. Különben nem képes feldolgozni. A nyelvet, amelyet megért, BASICnek nevezik. A gép udvariasságból már a bekapcsolásnál megnevezi az anyanyelvét. (Lapozzuk csak fel az első fejezetet!)

Próbáljuk ki, hogyan lehet vele megértetni a COMMODORE szót. Takarítsuk ki a képernyőt, aztán gépeljük be - nagyon vigyázva, hogy ne tévesszük el - ezt a sort:

```
PRINT "COMMODORE"
```

(az idézőjelet a 2-es számjegy billentyűjén találjuk meg, a SHIFTtel hívható elő), utána pedig nyomjuk meg a RETURN gombot!

A képernyőn ez jelenik meg:

```
************************************
*PRINT "COMMODORE"                 *
*  COMMODORE                       *
*                                  *
*READY.                            *
*■                                 *
*                                  *
```

Megtanultuk, melyik az az utasítás, amelynek hatására a számítógép kiír bármit a képernyőre. Ez a szó a PRINT, és amit utána idézőjelbe teszünk, azt a gép hiánytalanul kiírja, amikor csak akarjuk.

Próbáljuk is ki: menjünk rá a cursorral a gép által kiírt COMMODORE szó első betűjére, és töröljük ki a a sort a szóközbillentyű segítségével. Utána vigyük fel a cursort egy sorral, és nyomjuk meg újból a RETURN billentyűt. A gép ismét kiírja a szót, és ez akárhányszor megismételhető - a számítógép türelme végtelen.

Újabb kísérlet következik. Takarítsuk ki a képernyőt, és gépeljük be a következőt:

```
PRINT "
```

Az idézőjel után nyomjuk meg a CTRL gombot, és vele együtt a 2-es számjegy billentyűjét!

Aki tudja már, hogy ilyenkor a cursor színe fehérre változik, azt meglepetés éri. A cursor színe ugyanis marad, amilyen volt, viszont az idézőjel után megjelenik egy negatív E betű. Ezt még sosem tapasztaltuk, de egyelőre ne törődjünk vele; mintha mi sem történt volna, gépeljük be a nevünket, és amikor a végére értünk, ismét irjunk egy idézőjelet! A képernyőnk ilyen lesz ekkor:

```
***************************
* PRINT "ESAJAT NEVUNK"   *
*                         *
*                         *
*                         *
*                         *
*                         *
```

Most nyomjuk le a RETURN billentyűt!

```
***************************
*PRINT "ESAJAT NEVUNK"    *
*  SAJAT NEVUNK           *
*                         *
*READY.                   *
*█                        *
*                         *
```

A képernyőn megjelent a nevünk, mint az előbb a COMMODORE szó, de FEHÉR színben, és utána a READY meg a cursor is fehér. Sőt, mostantól minden, amit gépelünk, fehérben jelenik meg, mindaddig, amig a cursor színét át nem állítjuk.

Akár észrevettük, akár nem, nagy diadalban volt részünk: sikerült utasítást adnunk a gépnek, méghozzá olyat, amelyet azonnal megértett és végre is hajtott. Ez tulajdonképpen már majdnem programozás volt: a program ugyanis nem más, mint ilyen és hasonló utasítások sorozata, azzal a kikötéssel megtoldva, hogy a gép milyen sorrendben hajtsa végre ezeket az utasításokat.

Miből is állt ez az utasítás?

Először is a PRINT szóból, amely angolul azt jelenti: nyomtatni, de felszólító módú igének is felfogható: NYOMTASS! E szóval hoztuk a számítógép

tudomására, hogy a következő lépésben valamit ki kell írnia a képernyőre.

Másodszor: az idézőjelből. Ez a PRINT utasítások után olyankor következik, amikor a gépnek változatlan formában kell kiírnia azt, ami az idézőjelen belül van - jelen esetben a saját nevünket.

Hogyhogy változatlan formában? Hiszen ott van az idézőjelen belül egy negatív E betű, amelyet semmilyen formában nem írt ki a gép! Nos, igen; az E betűt nem írta ki, viszont a cursor (és vele együtt az írás) színét fehérre változtatta. A cursorra vonatkozó utasításokat, ideértve a színét és a haladási irányát is, idézőjelen belül adjuk meg a gépnek. Ezzel az információval egyelőre nem sokat tudunk kezdeni, de később a gépnek ez a tulajdonsága nagyon jól jön meg.

Harmadszor: a saját nevünkből állt az utasítás. Ehelyett bármit beírhattunk volna, tetszésünk szerint, a gép ugyanúgy kiírta volna. Ez a része az utasításnak bármikor bármivé változtatható. (Angolul ezeket a szövegeket stringnek nevezik - mi is ezt a szót fogjuk használni rájuk. Ez egyébként zsineget, spárgát jelent, talán azért, mert a gép az idézőjelen belül álló betűket, jeleket úgy kezeli, mintha fel lennének fűzve egy zsinegre - tehát összefüggő egészként, mint egy füzért.)

Most egy kicsit elkanyarodunk az eddig követett útvonalról, de csak látszólag.

A számítógépet nyilvánvalóan azért nevezik így, mert számításokat lehet végezni vele. Mi meg most már ki tudja, mióta, csak rajzolgatunk, irkálunk vele, de még egy fia számítást sem csináltunk. Most már talán nem tűnik olyan ijesztőnek a gép ahhoz, hogy egy kis matekot is bevezessünk, csak úgy, a játék kedvéért.

Akinek van zsebszámológépe, ne vegye alapul annak a kezelési módját. A COMMODORE 64-gyel másképp kell bánni.

Számítsuk ki először, mennyi kétszer kettő! Írjuk fel a képernyőre:

```
2*2
```

Figyelem! A szorzás jele a BASIC nyelvben nem x, nem is a pont, hanem a csillag, amely a második sorban jobbról a harmadik billentyűn található!

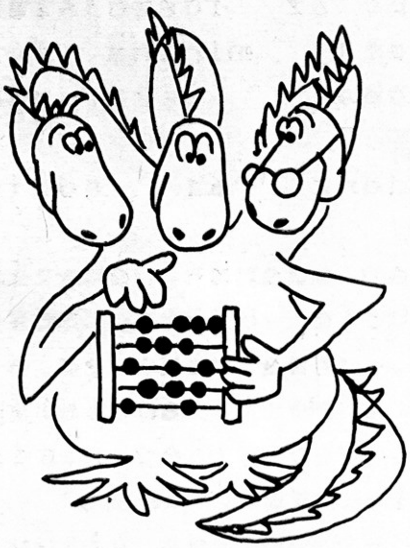
És most mi legyen? Az egyenlőségjelet hiába írjuk be, ettől a gép még nem írja ki az eredményt. Próbáljuk meg megnyomni a RETURN gombot! Semmi sem történik! A cursor a második sor elejére ugrik, de a 2*2 eredménye sehol sem jelenik meg.

Ne csigázzuk tovább a kíváncsiságot; eláruljuk, mi a baj. Nem mondtuk meg a számítógépnek, hogy mit akarunk tőle, ezért nem válaszolhatott. KÉRDEZZÜK MEG most tőle, mennyi 2*2! Gépeljük be a következő sort:

```
? 2*2
```

(Ha akarjuk, ezt értelmezhetjük mondatnak is: "Mennyi kétszer kettő?") A kérdőjelet egyébként a legalsó sorban, a jobb oldali SHIFT billentyű mellett találjuk meg. Ha most megnyomjuk a RETURN billentyűt, a képernyőn ez jelenik meg:

```
*********************************
*?2*2                           *
* 4                             *
*READY.                         *
*■                              *
*                               *
*                               *
*                               *
```



Megvan hát végre a várva várt eredmény, a 4! (Valljuk be: azért sejtettük előre, hogy ennyi jön ki...)

Még egy kísérlet:

```
? 9+5
```

Ha megnyomjuk a RETURN billentyűt, az eredmény (14) nyomban megjelenik a következő sorban.

Nézzünk meg egy példát:

```
? 15/2
```

(A ferde vonal a hármas számjegy előtt a BASIC nyelvben

az osztás jele, és a jobb oldali SHIFT billentyű mellett találjuk meg.)

Nyomjuk meg a RETURNt, és ott az eredmény: 7.5 - amiből kiderül, hogy a BASIC nyelv - az angol szokásoknak megfelelően - tizedesvessző helyett tizedespontot használ.

Komplikáltabb számításokat is gyorsan, és ami a fő: egy lépésben végeztethetünk el a géppel. Például:

```
? 7+(53-14)/3
```

Ugye ezt már nem lenne könnyű fejben kiszámítani? Ha viszont megnyomjuk a RETURN gombot, azonnal megtudjuk a végeredményt: 20.

Hát akkor vajon mi történik, ha a következő sort gépeljük be?

```
? "7+(53-14)/3="7+(53-14)/3
```

Hajmeresztő jelhalmaz, de ha jobban megnézzük, kiderül, hogy ugyanazt a műveletet írtuk le, mint az előbb, de most kétszer: egyszer idézőjelben, egyszer kívüle, és az idézőjelen belül találunk egy egyenlőségjelet is. Tudjuk: az idézőjelbe tett betűket, jeleket, számokat a gép változatlan formában írja ki a képernyőre. Így lesz ez most is. A RETURN billentyű megnyomása után a következőt válaszolja a gép:

```
7+(53-14)/3=20
```

Ebből kiderül, hogy ugyanazt a szám- és betűcsoportot a gép képes volt kétféleképpen kezelni: az idézőjelen belül szövegként, az idézőjelen kívül pedig matematikai műveletként.

De álljunk csak meg egy percre! Itt valami csalás van! Mostanáig úgy tudtuk, hogy a szövegek - stringek - kiírásához PRINT szóval kell utasítást adni a számítógépnek! Hol van itt ez a PRINT utasítás?

Szíves elnézést kérünk, de egy fontos információt mostanáig visszatartottunk. Íme: a PRINT utasítás helyett a COMMODORE 64 számítógépen kérdőjelet lehet használni. A gép ezt pontosan úgy értelmezi, mintha az egész PRINT szót begépeltük volna.

Tehát ha azt írjuk:

? "SAJÁT NEVÜNK"

az ugyanolyan, mintha azt írtuk volna:

PRINT "SAJÁT NEVÜNK"

és fordítva:

PRINT "7+(53-14)/3="7+(53-14)/3

ugyanaz, mint

? "7+(53-14)/3="7+(53-14)/3

Sokkal kényelmesebbé válik így a gép használata; az ötbetűs szó helyett elég egyetlen jelet leírni. (A gépnek ezt a tulajdonságát eddig azért titkoltuk el, mert a matekot lehetőleg jó sokáig kerülni akartuk, márpedig a kérdőjel használata logikailag akkor érthető, ha egy számítás végeredményét k é r d e z z ü k a géptől. Most viszont annál jobban lehet örülni az új lehetőségnek...)

Most már semmi sem tarthat vissza attól, hogy megírjuk az első programunkat. Gondosan, betűről betűre gépeljük be az itt következőket:

```
1 PRINT"COMMODORE"
2 PRINT"O"
3 PRINT"M"
4 PRINT"M"
5 PRINT"O"
6 PRINT"D"
7 PRINT"O"
8 PRINT"R"
9 PRINT"E"
```

Ne felejtsük el minden sor végén megnyomni a RETURN billentyűt!

Ez, akár hisszük, akár nem, már egy számítógépre írt program. Le is lehet futtatni, ha akarjuk. Takarítsuk le a képernyőt!

(Tessék? Takarítsuk le? De hát az első programunk, alig gépeltük be... Jó, jó, csak nyugalom. Attól, hogy a képernyőn nem látszik, a számítógép memóriajában még megvan a program. Nyomjuk csak meg egész nyugodtan a SHIFT és a CLR/HOME billentyűt.)

Jó. A képernyő üres, a cursor alaphelyzetben van. Lehet futtatni a programot.

Azt, hogy "fuss!", angolul úgy mondják: run. Így mondják a BASIC nyelvben is. Írjuk hát be azt a szót, hogy RUN, és utána nyomjuk meg a RETURN billentyűt!

```
**********************
*RUN                 *
*COMMODORE           *
*O                   *
*M                   *
*M                   *
*O                   *
*D                   *
*O                   *
*R                   *
*E                   *
*                    *
*READY.              *
*■                   *
```

Első programunk tehát abból állt, hogy autogramot kértünk a géptől.

Ebből már az is kiderül, hogy mit jelentenek azok a számok, amelyeket a program sorainak elejére írtunk. A számítógép sorra veszi ezeket, a legalacsonyabb számtól a legmagasabbig, és sorban végrehajtja őket, majd a legmagasabb sorszámú utasítás végrehajtása után leáll, és kiírja a READY szót. Ettől a szokásától csak külön kérésre hajlandó eltérni.

Töröljük le a képernyőt ismét, és adjuk ki újból a RUN parancsot! A gép megint kiírja kétszer a COMMODORE szót, ugyanúgy, ahogyan az előbb, és ezt mindaddig megteszi, amíg a gépet ki nem kapcsoljuk, a programot ki nem töröljük a memóriából – vagy át nem írjuk.

Hogyan lehet átírni? Ahhoz természetesen látni kell a programot. Hogy megnézhessük ismét, új parancsot

kell adni a gépnek, éspedig ezt: LIST. Ennek a magyar "liszt" szóhoz semmi köze sincs, bár ugyanúgy kell kiejteni. Azonos azonban a "lista" szavunkkal - végső soron mindkettő a latin nyelvből ered. (A magyarba közvetlenül került át, a BASICbe az angolon keresztül.)

Töröljük le a képernyőt, és írjuk be: LIST, majd nyomjuk le a RETURN gombot!

```
****************************
*LIST                      *
*1 PRINT"COMMODORE"        *
*2 PRINT"O"                *
*3 PRINT"M"                *
*4 PRINT"M"                *
*5 PRINT"O"                *
*6 PRINT"D"                *
*7 PRINT"O"                *
*8 PRINT"R"                *
*9 PRINT"E"                *
*                          *
*READY.                    *
*■                         *
```

Tessék, ismét itt a programunk, de nem pontosan ugyanúgy, ahogyan beírtuk. A kérdőjelek helyére a számítógép magától beírta a PRINT szót, azért, hogy a program listája áttekinthetőbb legyen.

Lehet most már változtatni a programon. Bármelyik betűt kicserélhetjük például az idezojeleken belül, ha odamegyünk a cursorral (de ha befejeztük egy sor javítását, sose felejtsük el megnyomni a RETURNt!), sőt még a sorrendet is megvaltoztathatjuk, ha átírjuk a program sorszamait, például a 4-es szam helyere 9-est ütünk, a 9-esre pedig 4-est (ekkor is meg kell nyomni a RETURN billentyűt!). Ha ezt tesszük, a szamitogep újrarendezi a sorokat, és ismét felallitja a növekvo számsorrendet. A fenti peldaban, ha a 4-es sort a 9-essel cseréljük ki, és lefuttatjuk a programot, ezt a szót fogja a függőleges sorban kiírni: C - O - M - E - O - D - O - R - M, aminek persze a világon semmi értelme sincs.

Van értelme viszont annak a szónak, hogy MODOR. Hogyan lehetne rábírni a számítógépet, hogy ezt a szót írja ki függőlegesen? A MODOR szó benne van

abban, hogy COMMODORE, csak az elejéről kellene elhagyni azt, hogy COM, a végéről meg az E betűt.

Ha az elöbb felcseréltük a 4-es és a 9-es sort, most cseréljük vissza: nem kell mást tennünk, mint hogy a cursorral rámegyünk a 4-es számra, 9-est ütünk a helyére, aztán megnyomjuk a RETURNt, majd a 9-es szám helyére 4-est ütünk, és ismét RETURN. Ha ezután kilistáztatjuk a programot (LIST + RETURN), akkor a függőleges COMMODORE szó ismét helyreáll.

Ki kell tehát törölni a programból az 1,2,3 és 9 számú sort. Mi sem egyszerűbb. Menjünk le a cursorral a lista utolsó sora utáni sorba, és nyomjuk meg az 1-es számjegy billentyűjét, utána pedig a RETURNt. A cursor lejjebb ugrott egy sorral. Írjunk ide egy 2-est, aztán megint RETURN; a következő sorba egy 3-ast írjunk, az azutániba pedig egy 9-est. A számítógép ilyenkor, ha csak sorszámot táplálunk be neki, az e számon található programsort kitörli, "elfelejti". Ha most újra kiíratjuk a listát, ilyen lesz a képernyőnk:

```
****************************
*LIST                      *
*4 PRINT"M"                *
*5 PRINT"O"                *
*6 PRINT"D"                *
*7 PRINT"O"                *
*8 PRINT"R"                *
*                          *
*READY.                    *
*■                         *
```

Futtassuk le a programot, és látni fogjuk, hogy működik. Az, hogy most nincs 1-es számú sor, nem zavarja a gépet; ő a legalacsonyabb sorszámúnál kezdi, és a legmagasabbnál fejezi be.

Ha megint listát kérünk a géptől, és a 6-os számú sorban a D betűt átírjuk N-re, akkor a futtatásnál MONOR nevet írja ki a gép. Meg lehet-e oldani, hogy MONOR helyett most azt írja: MONITOR?

No, ez nem könnyű. Két sort kellene beszúrni a 6-os és a 7-es közé, amit sajnos nem lehet. A programsorok sorszámának mindig egész számnak kell lennie. "Hat és fëledik" sor nem létezik.

Mit lehet tenni?

Helyet kell teremteni két uj sornak. Menjünk fel a programlistára a cursorral, es minden sorszam után (a PRINT szó elé) írjunk egy nullát! (Ne nagy O betűt!) Az, hogy a sorszámok és a PRINT utasítás között nem marad szóköz, ne zavarjon - a gép a következő listázásnál majd odateszi.

A 4-es sorból tehát 40-est csinaltunk, az 5-ösből 50-est, és így tovább. A számitogép tudomasul vett öt új programsort, és ha ujbol listat kerünk, ezt fogja kiírni:

```
****************************
*LIST                      *
*4 PRINT"M"                *
*5 PRINT"O"                *
*6 PRINT"N"                *
*7 PRINT"O"                *
*8 PRINT"R"                *
*40 PRINT"M"               *
*50 PRINT"O"               *
*60 PRINT"N"               *
*70 PRINT"O"               *
*80 PRINT"R"               *
*                          *
*READY.                    *
*■                         *
```

Most már bőven van hely; a 60-as és a 70-es programsor közé akar kilenc új sort is beirhatunk. Nem kell mast tennünk, mint kitörölni a memoriabol a 4-tol 8-ig terjedő programsorokat (most, az elobb mondtuk el, hogy ezt hogyan kell; nem ismeteljuk meg), aztan beirhatjuk a két új sort (a lista aljara írjuk; a gep majd a helyére teszi):

```
61 ?"I"
62 ?"T"
```

Futtassuk le így is, hogy megbizonyosodjunk rola: mindent jol csinaltunk-e.

Igaz, belátjuk, már jó ideje mást se csinálunk, mint nyakra-főre függőleges sorokat íratunk a számítógéppel, és ez nem valami érdekfeszítő feladat. Ha a gép tudna unatkozni, már biztosan ásítana. Az is igaz viszont, hogy közben megtanultunk egy sor hasznos dolgot. Például azt, hogy

- a programsorokat sorszámozni kell;

- a sorszámozást tízesével érdemes csinálni, így könnyebb utólag új sorokat közbeiktatni;

- a sorokat úgy lehet kitörölni a memóriából, hogy az üres sorszámot ütjük be a RETURN billentyűvel;

- ha a sorszámot változatlanul hagyjuk, akkor a programsort tetszés szerint változtathatjuk;

- a futtatásra, listázásra vonatkozó parancsokat mindig üres sorba kell írni, és utána megnyomni a RETURN billentyűt - satöbbi.

Egy dolgot nem tudunk még: hogyan lehetne ettől az unalmas programtól megszabadulni? Hát például úgy, hogy fáradságot nem kímélve beírjuk minden sornak a sorszámát önmagában, így egyesével kitörölhetjük a sorokat. Egy ilyen hétsoros program esetében ez nem is tartana sokáig, de hosszabb, pár száz vagy pár ezer (!) soros program esetén napokig sem végeznénk vele. Ezért van a BASIC nyelvben egy igen kényelmes utasítás: NEW. (Angolul annyit tesz: új.) Ha ezt beírjuk egy üres sorba, és leütjük a RETURN gombot, a gép azon nyomban elfelejti az egész programot, és várja az újat (ezért is NEW a parancs).

Próbáljunk valami érdekesebbet: feleseljünk a számítógéppel! (Pontosabban: ő fog feleselni velünk.) Írjuk be pontosan, betűről betűre a következő programot:

```
10 ? EZ EGY ROSSZ SOR."
20 "EZ IS"
30 PRNT "EZ SE JOBB."
40 ??"KOMOLYAN MONDOM, HAGYD ABBA!"
```

Próbáljuk meg most ezt a programot lefuttatni! Amint a RUN szó után megnyomjuk a RETURNt, a gép azonnal ránk pirít:

```
?SYNTAX ERROR IN  10
READY.
■
```

Próbálkozzunk újra. Írjuk be:

```
RUN 20
```

(futtasd a huszadik sortól), és nézzük meg, mi történik.

```
?SYNTAX ERROR IN  20
READY.
■
```

És ez így fog menni végig: a gép minden sorban nyelvtani hibát talál. De hiszen ez is volt a célunk ezúttal: szándékosan rossz sorokat írtunk, csak azért, hogy most feladat következhessen:

1) Javítsuk ki a fenti négy programsorban az összes hibát úgy, hogy a programot le lehessen futtatni!

További feladatok:

2) Egyetlen sorban számitsuk ki, hány másodperc van egy szökőévben! (Gyengébbek kedvéért: a szökőév 366 napos.)

3) Írjuk ki a saját nevünket, mindegyik betut mas színnel!

4) Írjuk ki mind a tizenhat szín nevét, mindegyiket a saját színével! Ezt úgy elegáns megcsinálni, ha 16 soros programot írunk hozzá.

Mostanáig úgy használtuk a számítógépet, hogy beleírtunk ezt-azt, még igazi programot is, aztán, amikor láttuk, hogy működik, már el is dobtuk, kitöröltük a memóriából. Egész idő alatt lehetett azonban sejteni, hogy eljön majd az idő, amikor már olyasmit hozunk össze, amit kár lenne kidobni; munkánk eredményét később is használni akarjuk.

Márpedig a számítógép, ha kikapcsoljuk, mindent elfelejt, a legzseniálisabb programot is.

Persze mindenki tudja már, hogy az adatok, programok tárolására többféle lehetőséget is kidolgoztak a szakemberek. A programokat magnókazettán és hajlékony mágneslemezen (angolul: floppy disken) is tárolhatjuk, és később bármikor újból elővehetjük, használhatjuk, amíg le nem töröljük. A kazettás megoldás sokkal olcsóbb és ezért elterjedtebb, a mágneslemezes viszont gyorsabb és könnyebben kezelhető.

Vegyük elő jegyzeteinket az előző nap utolsó programjáról (ha megcsináltuk): arról, amelyik a színek nevét a megfelelő színnel írja ki a képernyőre. Ha szégyenszemre még nem írtuk volna meg, most innen gépeljük be:

```
10 ? "{CTRL+1}  FEKETE"
20 ? "{CTRL+2}  FEHER"
30 ? "{CTRL+3}  PIROS"
40 ? "{CTRL+4}  CIANKEK"
50 ? "{CTRL+5}  BIBOR"
60 ? "{CTRL+6}  ZOLD"
70 ? "{CTRL+7}  KEK"
80 ? "{CTRL+8}  SARGA"
90 ? "{C=+1}  NARANCS"
100 ? "{C=+2}  BARNA"
110 ? "{C=+3}  ROZSASZIN"
```

```
120 ? "{C=+4} SOTETSZURKE"
130 ? "{C=+5} KOZEPSZURKE"
140 ? "{C=+6} VILAGOSZOLD"
150 ? "{C=+7} VILAGOSKEK"
160 ? "{C=+8} VILAGOSSZURKE"
```

(A programban szereplő CTRL és C= a két segédbillentyű neve, s a zárójelben az a számjegyes billentyű áll mellettük, amelyet velük együtt kell megnyomni. A képernyőn ilyenkor mindenféle negativ grafikus jelek jelennek meg, de ezeket nem kell megtanulnunk; elég, ha a számítógép tudja. A magunk részéről a programlistákban a későbbiekben is ilyen kapcsos zárójelben fogjuk jelezni, ha két billentyűt egyszerre kell megnyomni. Jobb ez, mintha a képernyőre kerülő jelet irnánk ki, ahogyan sok tankönyv teszi, mert azokból a jelekből roppant nehéz megállapitani, melyik billentyűhöz tartoznak. Amikor a programot futtatjuk, meg fogjuk látni, hogy a 70-es sz. sor akár ha ott se lenne, a "KEK" szó nem jelenik meg a képernyőn. Ez persze azért van, mert a képernyő is kék színű, meg a betű is, ezért nem látszik. Ha azonban fölmegyünk a látszólag üres sorra a - most világosszürke - cursorral, és végiglépkedünk rajta, kiderül, hogy ott van a szó a helyén.)

Nos, elég az hozzá, hogy megvan a programunk, amelyet most ki fogunk menteni magnókazettára.

MAGNÓ

A magnót a számítógéphez csak egyfele modon lehet hozzákapcsolni; ha elolvastuk a kezelési útmutatót, ezt nem téveszthetjük el. Ezután az eljarás a következő:

1) Tegyünk egy üres (vagy letörölheto) kazettát a magnóba,

2) Tekerjük a szalagot a legelejére,

3) Gépeljük be a következő szöveget:

```
SAVE "ELSO PROGRAMOM"
```

aztán nyomjuk le a RETURNt.

Erre a gép válasza a következő lesz:

PRESS PLAY & RECORD ON TAPE

Eleget leveleztünk a számítógéppel idegen nyelven, ideje lefordítani, miről van szó.
A "SAVE" szó azt jelenti: megmenteni. A program "megmentése", nem nehéz kitalálni, a szalagra vagy lemezre írás. A gép a memóriában levő programot átmásolja a szalagra, de a memóriából nem törli ki, tehát újra futtatható.

A számítógép válasza csak ennyit jelent: "NYOMD MEG A MAGNÓN A 'FELVÉTEL' ÉS 'LEJÁTSZÁS' GOMBOT".



Fogadjunk szót a számítógépnek. Nyomjuk meg a magnón a két gombot. Ekkor a képernyőről minden szöveg, jel eltűnik, most folyik a program kimentése. Mivel rövid a program, nem sok idő telik el, és ismét feliratok jelennek meg.

```
***********************************
*SAVE "ELSO PROGRAMOM"            *
*PRESS PLAY & RECORD ON TAPE      *
*OK                               *
*SAVING ELSO PROGRAMOM            *
*                                 *
*READY.                           *
*■                                *
*                                 *
```

Szalagon van az első programunk; mostantól bármikor elővehetjük, betölthetjük a komputer memóriájába, és újra lefuttathatjuk.

Az ördög azonban nem alszik. Előfordulhat olykor, hogy áramingadozás történt a hálózatban, vagy bármi más közbejöhet, és a program nem pontosan ugyanúgy kerül a szalagra, ahogy a memóriába beírtuk. Célszerű addig ellenőrizni ezt, amíg a memóriában még az eredeti változat van. A számítógép össze is tudja hasonlítani a kettőt. Ennek módja a következő:

1. Nyomjuk meg a magnón a STOP gombot.

2. Tekerjük a szalagot a legelejére.

3. Gépeljük be: VERIFY (ez az angol szó azt jelenti: igazolni), és nyomjuk meg a RETURNt. A gép válasza a képernyőn: PRESS PLAY ON TAPE. (Nyomd meg a magnón a PLAY - lejátszás - gombot!)

4. Nyomjuk meg a magnón a PLAY (lejátszás) gombot! Ekkor minden eltűnik a képernyőről, majd, amikor ismét megjelenik, ezt látjuk:

```
**********************************
*VERIFY                          *
*PRESS PLAY ON TAPE              *
*OK                              *
*SEARCHING                       *
*FOUND ELSO PROGRAMOM            *
*                                *
*                                *
*                                *
```

(A géppel folytatott párbeszéd magyarul kb. így hangzana: "Igazold!" - "Akkor te meg nyomd meg a Lejátszás gombot a magnón!" - "Oké!" - "Keresek..." - "Megtaláltam az 'Első programom' címü programot.")

Itt a magnó megáll, a gép vár néhány másodpercig, aztán olvasni kezdi a szalagról a programot, és olvasás közben összehasonlítja azzal, amit begépeltünk. Ha ezt a várakozást meg akarjuk spórolni, nyomjuk meg a C= gombot a gépen, s az összeolvasás azonnal megkezdődik.

5. Ha magnó ujból megál., a számitógép ismét "OK" üzenettel mondja meg, hogy a program sikeresen átkerült a szalagra. Ilyenkor ki lehet venni a kazettát a magnóból, és ráírni, hogy milyen címü programok vannak rajta. Most, az első programunk felvételénél ez még nevetséges óvatoskodásnak tűnhet, de előbb-utóbb százával tároljuk majd a programokat, és a nyilvántartásuk nem lesz könnyű az ilyen feljegyzések nélkül.

6. Tegyünk egy merész kísérletet! Töröljük ki a memóriából a programot, és töltsük be ujból a szalagról! Most kiderül majd, hogy mindent jól csináltunk-e. Kitörölni, mint tudjuk, a NEW szó begépelésével kell, amire a gép READYvel válaszol.

Ezután írjuk be a következőket:

LOAD "ELSO PROGRAMOM"

A szalagnak természetesen az elején kell állnia, erről előre gondoskodjunk. (A LOAD szó - lódnak kell ejteni - annyit tesz: tölts!) A gép azt feleli (mint már korábban is):

PRESS PLAY ON TAPE

Ami, tudjuk már jól, azt jelenti, hogy meg kell nyomnunk a lejátszás gombot a magnón. Jó, nyomjuk meg. A magnó forogni kezd, a képernyőn semmi sem látszik, aztán jön az üzenet:

FOUND ELSO PROGRAMOM

Innentől egyszerű: várjunk egy kicsit, vagy, ha türelmetlenek vagyunk, nyomjuk meg a C= gombot, és kisvártatva megjelenik a READY üzenet. Ekkor a programunk ismét bent van a számítógép memóriájában, lehet futtatni.

MÁGNESLEMEZ

Ez már komoly dolog. A mágneslemezt nem lehet úgy használni, mint a magnókazettát, hogy csak úgy berakjuk a helyére, aztán gyerünk, felveszünk, visszaolvasunk, igazolunk, ahogy tetszik. Ahogyan a mágneslemezt megvesszük a boltban, úgy még nem lehet használni. Előbb formát kell adni neki, azaz, angolosan: "formattálni" kell.

A formattálás során minden lemez saját nevet és számot kap: például így hívhatják: "JATEKOK,99". Ez jelentheti azt, hogy száz lemez közül, amelyek mind játékot tartalmaznak, ez az utolsó. (Hogy miért nem az utolsó előtti? Hát mert egy gyakorlott komputeres, akinek már száz lemezre való játéka gyűlt össze, az biztosan nem 01-nél, hanem 00-nál kezdi a számozást... Van még kérdés?)

Szóval a saját céljainkra saját rendszer szerint formattálhatjuk a lemezeinket. Természetesen elnevezhetjük mindegyiket így is: "AZ EN LEMEZEM,XX", de hát ez rettentő nagy butaság volna. Inkább valami áttekinthető rendszert találjunk ki magunknak, hogy később is gyorsan meg tudjunk találni mindent, amit keresünk.

Tegyünk be egy új lemezt a meghajtó egységbe (drive-ba; ezt az angol szót drájvnak kell ejteni). Aztán pontosan, betűről betűre gépeljük be a következőket:

```
OPEN 15,8,15, "N:
```

a kettőspont után pedig írjuk be azt a nevet, amelyet a lemezünknek szánunk. Legyen ez egyelőre a saját nevünk.

```
OPEN 15,8,15, "N:SAJAT NEVUNK,01":CLOSE15
```

Egy dologra kell még vigyázni: a lemez neve legföljebb 16 karakter (betű, számjegy, grafikus jel) lehet, és ha szóköz van benne, az is külön karakternek számít. Ha tehát valakit Kiskunfélegyházi Sándor József Benedeknek hívnak, az csak a vezetéknevét használhatja lemeznévként, a többi egyszerűen nem fér rá. A név után következő két számjegy (példánkban 01) akármilyen két karakter lehet, de csak kettő. Ezt szaknyelven ID-számnak nevezik, ami az angol "identity" (azonosság) szó rövidítése.

Nomármost: ha ezt a sort begépeltük, olvassuk el figyelmesen, hogy minden betű, vessző, kettőspont, idézőjel stb. a helyén van-e, aztán nyomjuk meg a RETURNt!

A drive piros lámpája kigyullad, belülről duruzsolás hallatszik, majd rövid ideig valami éktelen kopácsolás hangja. Ettől ne ijedjünk meg; amikor lemezt formattálunk, ez mindig így fog történni. Jó néhány másodpercig (talán egy percig is) dolgozik a drive, amíg használatra előkészíti a lemezünket. Utána megáll. A READY már korábban is ott volt a képernyőn, de a lemez csak akkor kész, ha a drive leállt!

A formattálással nagyon vigyázzunk: egy lemez életében általában csak egyszer szabad megcsinálni. Ha ugyanis véletlenül - vagy szándékosan - újra

formattáljuk, minden, ami addig a lemezen volt, egyszer s mindenkorra odavész, soha többé nem lehet visszaszerezni, ugyanúgy, ahogyan egy letörölt magnószalagról sem.

Ellenőrizzük, jól csináltuk-e. Hívjuk be a lemez tartalomjegyzékét a gép memóriájába. A tartalomjegyzék (angolul: directory, ejtsd: dajrektori) jele a dollár ($), amelyet a 4-es számjegy billentyűjén találunk meg, és SHIFT-tel hívhatunk elő. Irjuk be tehát a következőket:

```
LOAD "$",8
```

és utána nyomjuk meg a RETURNt. (Aki figyelmesen olvasta idáig a könyvet, annak a LOAD szó ismerős, de azt is észreveheti, hogy amikor magnoról töltöttük be a programot, nem volt ott az a nyolcas számjegy. Igy igaz: a számitogep ettől a nyolcas szamjegytől tudja, hogy ne a magnóról, hanem a drive-ról töltse be a kért programot.)

Amíg a fenti bekezdést elolvastuk, a számítógép már be is hozta a tartalomjegyzéket, hiszen az ilyenkor még nagyon rövid; nem tartalmaz mást, mint a lemez nevét és az ID-számot. A képernyőn tehát ezt látjuk:

```
*********************************
*LOAD "$",8                     *
*SEARCHING FOR $                *
*LOADING                        *
*                               *
*READY.                         *
*■                              *
*                               *
```

A memóriában van tehat lemezünk tartalomjegyzéke, amelyet mindjárt el is olvashatunk. Gépeljük be a LIST szót (ugyanaz a LIST ez, mint amelyikkel a programjainkat listaztathatjuk ki), es nyomjuk meg a RETURNt.

```
****************************************
*LOAD "$",8                            *
*SEARCHING FOR $                       *
*LOADING                               *
*                                      *
*READY.                                *
*LIST                                  *
*0 "SAJAT NEVUNK       " 01 2A         *
*664 BLOCKS FREE.                      *
*                                      *
*READY.                                *
*█                                     *
*                                      *
```

Mit látunk ezen a listán? Először is egy nullát, aminek számunkra nincs jelentősége. Utána idézőjelben, negatív karakterekkel a saját nevünket látjuk - ez mostantól a lemezünk neve is. Ezt követi az ID-szám, már az idézőjelen kívül, majd két karakter, amellyel ugyancsak nem kell törődnünk.

Ezután egy olyan információ következik, amely már nekünk szól: 664 blokk szabad a lemezen: ennyi helyünk van programok felvételére. Hogy pontosan mennyi 664 blokk, azt akkor tudjuk meg, ha fölvesszük 16 soros programunkat lemezre, és megnézzük, az mennyi helyet foglal el.

Gépeljük be a NEW utasítást, hogy kitöröljük a tartalomjegyzéket a memóriából (ha ezt elfelejtjük, a directory összekeveredhet a programunkkal, amit mindjárt beírunk).

Lapozzunk e fejezet elejére, gépeljük be a programot, a biztonság kedvéért futtassuk is le, hogy jól működik-e, aztán írjuk a képernyőre a következőket:

```
SAVE "ELSO PROGRAMOM",8
```

(Látjuk, itt is szerepel a nyolcas, mint ezután is mindig, ha a drive-ot akarjuk dolgoztatni.) Nyomjuk meg a RETURNt.

```
***********************************
*SAVE "ELSO PROGRAMOM",8          *
*SAVING ELSO PROGRAMOM            *
*                                 *
*READY.                           *
*■                                *
*                                 *
```

Ezt is lehet a VERIFY utasítással ellenőrizni (VERIFY "ELSO PROGRAMOM",8), és utána kérjünk ismét tartalomjegyzéket a géptől!

```
***********************************
*LOAD "$",8                       *
*SEARCHING FOR $                  *
*LOADING                          *
*                                 *
*READY.                           *
*LIST                             *
*0 "SAJAT NEVUNK [illegible] 01 2A     *
*2 "ELSO PROGRAMOM"         PRG   *
*662 BLOCKS FREE.                 *
*                                 *
*READY.                           *
*■                                *
*                                 *
```

Kiderül hát, hogy eléggé pazarló módon bántunk a memóriával: két blokkot foglaltunk el a lemezen ezzel a programmal. Igaz ugyan, hogy még háromszázharmincegyszer fölvehetnénk ugyanezt a programot ugyanerre a lemezre, de azt is tudnunk kell, hogy ennél jóval rövidebben is meg tudjuk majd írni a programunkat, ha már ügyesebbek leszünk.

5. NAP

Emlékszünk, hogy egyszer, a második napon az volt a feladat: rajzoljunk kockás füzetlapot a képernyőre! Megmondtuk előre, hogy több megoldás lehetséges, de mindegyik magában foglalta azt a kellemetlen körülményt, hogy annyiszor kell lenyomni valamelyik grafikus jel billentyűjét (leginkább a + jelet, SHIFTtel), ahány jelet akarunk a képernyőre írni. Ez pedig több száz gombnyomást jelent. Nagyon csalódnánk a számítógépben, ha ennél egyszerűbb megoldást nem tudna. Aki még soha nem próbálkozott programozással, az aligha jött rá a leggyorsabb módjára: írjunk erre egyetlen kétsoros programot!

```
10 ?"+"
20 GOTO 10
RUN
```



Mit látunk? A képernyő bal szélén rettentő gyorsasággal egymás alá íródnak a + jelek, és ha nem csinálunk semmit, ez így folytatódik, amíg ránk nem esteledik, és ki nem kapcsoljuk a gépet.

Csináljunk hát valamit: nyomjuk meg a RUN/STOP feliratú gombot. A képernyőn megáll a + jelek szédült futása, és a legalsó sorokban ezt olvashatjuk:

```
BREAK IN 20 (vagy BREAK IN 10)
READY.
```

Szokás szerint elmagyarázzuk, mi mit jelent. A 10-es programsorban nincs újdonság. A kérdőjel a PRINT szó rövidítese, a pluszhoz hasonló, de annál nagyobb grafikus jel pedig azért van idézőjelek között, mert szöveges változó (azaz string - emlékszünk még?).

Vadonatuj elem van azonban a 20-as sorban. GOTO - mi a manó ez?

A BASICben ez egy utasítás, de az angol nyelven két szó: go to... Utána okvetlenül még valaminek kell állnia, mert szó szerint ennyit jelent: "menj ...-hez". A három pont helyébe valamit be kell írni, hogy mondat legyen. Mi azt írtuk be: 10. Ez annyit jelent, ebben az esetben, hogy "menj a 10-es sorra". A tízes sorban pedig ugyebár az áll, hogy írj ki egy + jelet.

A számitógep - mint mondtuk - növekvő számsorrendben veszi és hajtja vegre a program sorait, és ettől csak külön kérésre hajlandó eltérni. Nos, a GOTO eppen egy ilyen külön kérés. Ha ez nem volna itt, akkor a gép leírt volna egy + jelet, és utána READY-vel jelezte volna, hogy a program vegere ert. Így azonban a 20-as sorban visszafordult, és ismet végrehajtotta a 10-es sor utasítását.

Ha kilistáztatnánk a programot, és hozzáírnánk egy 30-as sort, például így:

```
10 ? "+"
20 GOTO 10
30 ? "SAJAT NEVUNK"
```

és lefuttatnánk, ugyanazt tapasztalnánk, mint az előbb: csak a + jelek sorakoznának egymás alá, a nevünket nem látnánk meg a képernyőn. A gép ugyanis nem jut el a 30-as sorig: a 20-asból visszafordul, és így folytatja, amíg van benne áram.

Amikor a RUN/STOP gombot megnyomtuk, megállitottuk a program futását, mert végtelen körbe kergettük a gépet, ahonnan magától nem tudott volna kiszabadulni. Ekkor azt üzente vissza: BREAK IN 20. A BREAK szó itt törést, szakadást jelent, a gép tehát visszajelezte, hogy megszakitottuk a program futását.

Szép, szép, de ez még nem kockás füzetlap, csak annak a bal széle. A gép, valahányszor a 10-es sorhoz ért a program, új sort kezdett a képernyőn,

ezért kerültek egymás alá a + jelek. Hogyan lesz ebből egy sorban 40 jel?

Szinte hihetetlenül egyszerűen. Takarítsuk ki a képernyőt (SHIFT + CLR/HOME), és listáztassuk ki a programunkat. Utána lépkedjünk fel a cursorral a 10-es sorba, és a legvégére, mindjárt az idézőjel után írjunk egy pontosvesszőt!

```
10 ? "+";
20 GOTO 10
```

És kész! Ha ezt most lefuttatjuk, a számítógép gyönyörű négyzethálót rajzol a képernyőre; fáradhatatlanul, amíg mi meg nem únjuk, es újból meg nem nyomjuk a STOP gombot. Akkor persze megint megkapjuk az üzenetet: BREAK IN 20.

Jegyezzük meg: a pontosvessző (az idézőjelen kívül) a BASIC nyelvben azt eredményezi, hogy a PRINT utasítás után a gép nem kezd új sort, hanem közvetlenül az előzőleg kiírt string után folytatja az írást.

Jó ezt tudni, és előre felkészülni rá. Néha ebből zavarok is támadhatnak. Például...

Listázzuk ki a programot, és javítsuk ki a 10-es sort, hogy a string a saját nevünk legyen!

```
10 ? "SAJAT NEVUNK";
20 GOTO 10
```

Mindent értünk: az idézőjelek között a saját nevünk áll, ezért a gép most azt fogja ilyen szorgosan kiírogatni, amíg le nem állítjuk.

Futtassuk le a programot. A képernyő megtelik szöveggel, ami már futás közben is emlékeztet a saját nevünkre, de valahogy fura. Ha leállítjuk a STOP gombbal, meg is látjuk, hogy miért.

```
****************************************
*SAJAT NEVUNKSAJAT NEVUNKSAJAT NEVUNKSAJA*
*T NEVUNKSAJAT NEVUNKSAJAT NEVUNKSAJAT NE*
*VUNKSAJAT NEVUNKSAJAT NEVUNKSAJAT NEVUNK*
*SAJAT NEVUNKSAJAT NEVUNKSAJAT NEVUNKSAJA*
*T NEVUNKSAJAT NEVUNKSAJAT NEVUNKSAJAT NE*
*VUNKSAJAT NEVUNKSAJAT NEVUNKSAJAT NEVUNK*
*                                        *
*BREAK IN 20                             *
*READY.                                  *
*■                                       *
```

Azonnal látható, hogy mi a hiba: csak az első és a második szó között hagytunk szóközt, a második és az első között nem (!). Azt már az első napon láttuk, hogy hogyan lehet szóközt létrehozni olyan sorban, ahol nincs, most frissítsuk fel az akkori tudásunkat!

```
10 ? "SAJAT NEVUNK";
20 GOTO 10
```

Menjünk fel a cursorral a 10-es sor végére, állítsuk rá a második idézőjelre, aztán nyomjuk meg a SHIFTet és az INST/DEL billentyűt. Az idézőjel meg a pontosvessző jobbra húzódik, és egy hézag keletkezik a cursor helyén. Most megnyomjuk a RETURNt, és a gép mostantól ezt tekinti a programunk 10-es számú sorának.

Ha lefuttatjuk, azonnal látszik a különbség:

```
****************************************
*SAJAT NEVUNK SAJAT NEVUNK SAJAT NEVUNK S*
*AJAT NEVUNK SAJAT NEVUNK SAJAT NEVUNK SA*
*JAT NEVUNK SAJAT NEVUNK SAJAT NEVUNK SAJ*
*AT NEVUNK SAJAT NEVUNK SAJAT NEVUNK SAJA*
*T NEVUNK SAJAT NEVUNK SAJAT NEVUNK SAJAT*
* NEVUNK SAJAT NEVUNK SAJAT NEVUNK SAJAT *
*                                        *
*BREAK IN 20                             *
*READY.                                  *
*■                                       *
```

Kísérletképpen írjunk egy olyan programot, amely a számítógép grafikai jeleiből igazi kis képecskét rajzol a képernyőre. Ehhez előljáróban egy pár dolgot még el

kell mondani. Többek között egy fontos BASIC utasítást, éspedig azt, hogy REM.

Ez az angol "remark" (megjegyzés) szó rövidítése, és a szerepe nagyon érdekes: ami a programsorban a REM után következik, azzal a számítógép egyáltalán nem törődik. A REM csak a programozó számára hordoz üzenetet, magyarázatot, a számítógép úgy átszalad fölötte, mintha ott se lenne. Ha a REM a programsor elején áll, akkor az egész sort kihagyja a gép a futtatásnál, ha viszont a sor végén, akkor az előtte álló programutasítást végrehajtja, és amikor a REMmel találkozik, akkor ugrik át a következő sorra. Ne felejtsük el: ebben az esetben a REM elé kettőspontot kell tenni, hogy elválasszuk az előtte álló utasításoktól.

Ez így eléggé homályosnak tűnhet, de a gyakorlatban mindjárt megvilágosodik. Lássuk a programot, amely egy babafejet fog kirajzolni a képernyőre!

```
10 PRINT"  ●●●●●●●"     : REM SHIFT+Q
20 PRINT" ●●●●●●●●●"
30 PRINT"●●●      ●●●"
40 PRINT"●●        ●●"
50 PRINT"●● ▔▔ ▔▔ ●●"   : REM C=+U
60 PRINT"●● ╭╮  ╭╮ ●●"  : REM SHIFT+U : SHIFT+I
70 PRINT"●● ╰╯  ╰╯ ●●"  : REM SHIFT+J : SHIFT+K
80 PRINT"●●    ○   ●●"  : REM SHIFT+W
90 PRINT"●●        ●●"
100 PRINT"●   ╰─╯    ●": REM SHIFT+J : SHIFT+* : SHIFT+L
110 PRINT"  ╲      ╱"  : REM SHIFT+M : SHIFT+N
```

Ha lefuttatjuk, a baba vigyori képe tűnik fel a képernyőn. Figyeljük meg: ha a képernyő üres, és a bal felső sarokba gépeljük a RUN szót, akkor a kép szebben, gyorsabban jön elő, mint ha – mondjuk – a program listája alá írjuk a RUNt. Ebben az utóbbi esetben előfordulhat, hogy a fölfelé haladó kép nem is tolja le a képernyőről a teljes listát, hanem valahol alatta rajzolódik ki; röviden: csúnya a képernyő.

Tudjuk már régóta, hogyan lehet a képernyőt kitakarítani, de sokkal elegánsabb volna, ha a program maga végezné ezt el helyettünk. Erre is van mód.

A program legelejére kell írnunk, hogy "töröld tisztára a képernyőt!", aminek szép és elegáns módja van a COMMODORE-on.

Programunk első sora a 10-es számot viseli, tehát (0 és 9 között) tíz különböző sorszámot adhatunk annak a sornak, amely a takarítási feladatot végzi. Legyen ez a 0. Gépeljük be az alábbi sort (mindegy, hogy van-e valami a képernyőn, vagy nincs; az a lényeg, hogy az a sor legyen üres, amelyikbe írunk):

```
0 ? " SHIFT+CLR/HOME "
```

Érthető, ugye? Eddig mi magunk nyomtuk meg a SHIFT és a CLR/HOME billentyűt, most ezt a programra bízzuk. (A 0 számú sorban az idézőjelek között megint egy negatív karaktert látunk: azt a szívet, amely az S betű billentyűjén található.)

Futtassuk le a programot. A babafej most már üres képernyőre rajzolódik ki.

Csinosítsuk ki még jobban a babát: rajzolódjon ki középszürke keretben, fehér alapon fekete jelekkel! Ehhez elő kell vennünk a pár nappal korábban található színkód-jegyzéket. A keretszínt a POKE 53280, utasítással tudjuk változtatni, a háttérszínt a POKE 53281,-vel, és a vessző után a megfelelő szín kódszáma kerül. A középszürke kódszáma 13, a fehéré pedig 1. A cursort is feketére kell változtatni - ez összesen 3 utasítás, amit a nullás és a tízes programsor közé kell beírni. Például így:

```
3 POKE 53280,13
5 POKE 53281,1
7 ? " CTRL+1 "
```

Futtassuk le a programot, és ha hibátlannak találjuk, mentsük is ki mindjárt kazettára, úgy, ahogy az előbb. (A programnak, tudjuk, nevet kell adni; nevezzük így: "BABA".)

Ne szaporítsuk a szót azzal, hogy hogyan kell magnóra kimenteni: tessék visszalapozni, ha valaki elfelejtette. Inkább térjünk rá a következő programra.

```
0 REM COMMODORE
10 POKE53280,13
20 POKE53281,8
30 PRINT"□":REM CLEAR HOME
40 PRINT"■":REM CTRL+3
50 PRINT"▒▒▒▒▒▒▒":REM CRSR LE
60 PRINT"████████████████C":REM CRSR JOBBRA
70 PRINT"█████████████████D"
80 PRINT"██████████████████M"
90 PRINT"███████████████████M"
100 PRINT"████████████████████D"
110 PRINT"█████████████████████D"
120 PRINT"██████████████████████D"
130 PRINT"███████████████████████R"
140 PRINT"████████████████████████E"
150 PRINT"███████████████████████R"
160 PRINT"██████████████████████D"
170 PRINT"█████████████████████D"
180 PRINT"████████████████████D"
190 PRINT"███████████████████M"
200 PRINT"██████████████████M"
210 PRINT"█████████████████D"
220 GOTO 60
```

A cursorbillentyű nyomogatásakor különféle negatív karakterek jelennek meg a képernyőn – a lefelé mozgatásnál egy negatív Q betű, a jobbra mozgatásnál egy szögletes zárójel. Ez normális esemény; csak akkor javítsuk, ha nem ezt tapasztaljuk.

Adjunk most egy RUNt – a képernyőn cikcakkos vonalban a COMMODORE szó fut felfelé – amíg meg nem állítjuk.

Tegyük el ezt a programot is szalagon (vagy, aki teheti, lemezen), mert a következő napon megint használni fogjuk. Legyen a program neve (ehhez nem kell nagy leleményesség): COMMODORE.

Az előző napon nem beszéltünk a levegőbe: kezdjük mindjárt azzal, hogy betöltjük a COMMODORE nevű programunkat.

Emlékszünk rá, ez egy szép cikcakkos COMMODORE feliratot csinált a képernyőn, és sohasem hagyta abba, mert az utolsó sorból visszaküldtük az elejére. Van azonban olyan lehetőség is, hogy csak egy bizonyos, általunk meghatározott számú alkalommal menjen vissza, utána pedig hagyja abba.

Ezt a FOR... NEXT ciklussal érhetjük el.

Most bajban van e sorok írója: ennek a kifejezésnek a pontos magyar fordításával alighanem adós marad. Mindenesetre megpróbáljuk körülírni, de csak akkor, ha már a programunkat átalakítottuk.

```
0 REM COMMODORE
10 POKE53280,13
20 POKE53281,8
30 PRINT"█":REM CLEAR HOME
40 PRINT"█":REM CTRL+3
50 PRINT"██████":REM CRSR LE
60 PRINT"███████████████C":REM CRSR JOBBRA
70 PRINT"████████████████O"
80 PRINT"█████████████████M"
90 PRINT"██████████████████M"
100 PRINT"███████████████████O"
110 PRINT"████████████████████D"
120 PRINT"█████████████████████O"
130 PRINT"██████████████████████R"
140 PRINT"███████████████████████E"
150 PRINT"██████████████████████R"
160 PRINT"█████████████████████O"
170 PRINT"████████████████████D"
```

```
180 PRINT"████████████████████O"
190 PRINT"███████████████████M"
200 PRINT"██████████████████M"
210 PRINT"█████████████████O"
220 GOTO 60
```

Így fest a program most, ha kilistázzuk. Írjunk bele két sort, így:

```
45 FOR N=1 TO 10
220 NEXT N
```

45-ös sorunk még nem volt, ezért azt a számítógép a 40-es és az 50-es közé fogja beszúrni, a régi 220-as sort pedig egyszerűen eldobja, és a helyére beteszi az újat. (Aggodalomra semmi ok; a régi program attól még megvan a kazettán!)

Mit csinál ez a két sor a programunkban? Megnevez egy változót (egy olyan fogalmat, amelyhez különböző számértékek tartozhatnak): az N-et, és megmondja a számítógépnek, hogy ennek az N-nek az értéke 1-től 10-ig terjedhet. (Azt nem kell külön megmondani, hogy csak egész szám lehet, a gép ezt magától tudja.) A 45-ös sor tehát körülbelül így fordítható le emberi magyar nyelvre: "N minden értékére hajtsd végre a FOR és a NEXT közötti utasításokat, miközben az N-et növelgeted 1-től 10-ig." A NEXT utasítás hatására a számítógép visszaugrik arra a sorra, amelyikben a FOR áll. A NEXT angolul annyit jelent: "következő", tehát az N fölveszi a következő értéket, egészen a FOR sorban megjelölt határig, ami a példánkban 10. (Ez a FOR... NEXT is olyan "külön kérés", amilyen a GOTO, amellyel az előző fejezetben foglalkoztunk.) Az N értéket egyesével végigszámolja a gép, tehát összesen kilencszer fordul vissza a NEXTtől, így fog tízszer végigfutni a programon, utána pedig szépen megáll, és kiírja: READY.

Tudni kell: N helyett írhattunk volna I betűt, vagy G-t, vagy amit akarunk, a lényeg az, hogy betű legyen, ne szám, mert azt SYNTAX ERRORnak, azaz nyelvtani hibának tekinti a gép.

Írjunk egy egyszerűbb programot:

```
10 PRINT"🗋" : REM SHIFT+CLR/HOME
20 FOR F=1 TO 10
30 PRINT "SAJAT NEVUNK"
40 NEXT F
```

Ha lefuttatjuk a programot, tízszer fog egymás alatt megjelenni a nevünk, s alatta a READY szó.

Számoljuk meg, valóban tízszer írta-e ki a gép! Biztos, hogy igen, de jobb lenne, ha a számozást is a képernyőn látnánk. Egyszerű: a 30-as sorban meg kell mondani a gépnek, hogy ne csak a nevünket, hanem az F mindenkori értékét is írja ki. Hogyan kell ezt megmondani? Alakítsuk át a 30-as programsort így:

```
30 PRINT F;"SAJAT NEVUNK"
```

Így lefuttatva a programot, a képernyőn szépen 1-től 10-ig megszámozva ott fogjuk látni a nevünket.

Figyeljük meg: az F-et nem az idézőjelen belül írtuk, hanem eléje, hiszen az F értéke mindig változik, az idézőjelen belüli szöveget pedig a gép mindig változatlan formában írja ki – ezt is régóta tudjuk már.

Egy hasznos tanács: ha a program futása közben lenyomjuk a CTRL billentyűt, a képernyőn lassabban jönnek fel az új sorok, jobban lehet követni a program futását.

Most írjunk egy olyan programot, amelyik 2-től százig kiírja a páros számokat! Hogy hogyan kezdjünk hozzá, azt tudjuk:

```
10 PRINT"🗋"
20 FOR I=2 TO 100
30 PRINT I
40 NEXT I
```

Ez biztos. De az is biztos, hogy a számítógép így az összes számot kiírja 2 és 100 között, nemcsak a párosakat. Meg kell mondani neki, hogy minden másodikat hagyja ki, ezt pedig a még eddig nem ismert STEP utasítással lehet az értésére adni. (A STEP egyébként lépést jelent angolul, ami logikus is: 2 lépésenként

kell most a kiírást végrehajtania.) Programunk 20-as sorát alakítsuk át így:

```
20 FOR I=2 TO 100 STEP 2
```

Működik a program, ha lefuttatjuk, de megint a képernyő bal szélére szorítva, egymás alá írja a számokat. Próbáljunk pontosvesszőt tenni a 30-as sor végére!

```
30 PRINT I;
```

Ha most futtatjuk le programunkat, már ráfér egy képernyőre az összes szám, de még mindig nem elég áttekinthető: az első sor elején kis zűrzavar van. Hiába, úgy látszik, mániánk, hogy fontos információkat az utolsó pillanatig visszatartunk. Például azt is elhallgattuk mostanáig, hogy mire szolgál a BASIC nyelvben a vessző. De nem baj, mindjárt meglátjuk. Írjunk egy vesszőt a 30-as sorban a pontosvessző helyére!

```
30 PRINT I,
```

Futtassuk le most a programot! Nagyszerű, áttekinthető táblázatot kapunk:

```
*******************************************
*                                         *
* 2         4         6         8         *
* 10        12        14        16        *
* 18        20        22        24        *
* 26        28        30        32        *
* 34        36        38        40        *
* 42        44        46        48        *
```

És így tovább, és így tovább, egészen 100-ig.

Szóval: mit jelent a vessző? A képernyőn 24 sor fér el, és minden sorba 40 karakter (betű, jel, számjegy stb.) írható. Ha vesszőt teszünk a képernyőre kiírandó változó neve után, akkor a gép nem írja mindegyiket új sorba, és nem is közvetlenül az előző után, hanem tízes oszlopokra osztja a képernyőt, és minden PRINT utasítás után a következő ilyen tízes oszlop elejére ugrik, ott folytatja az írást.

(Jó, igaz, nem épp az elejére: mindig kihagy egy karakternyi helyet. Ez az egy karakter azonban az előjel számára kell: ha az előjel pozitív, a gép nem írja ki, ha viszont negatív, akkor mégis kell neki a hely. Aki nem hiszi, járjon utána: javítsa ki a program 20-as sorában a 2 előjelet, legyen I=-2, és kiderül, hogy igazat mondtunk.)

Írjunk most olyan programot, amely 0 és 100 között sorolja fel az összes néggyel osztható számot, de fordítva, a legmagasabbtól a legalacsonyabbig! Ez még jó matekosnak is nehézséget okozna, a számítógépnek azonban semmiség. Csak a programot kell jól megírni hozzá.

```
10 PRINT "□"
20 FOR Z=100 TO 0 STEP -4
30 PRINT Z,
40 NEXT Z
```

A program működik, pedig a lépésszám most negatív: ezt akkor használjuk, ha a FOR után álló változó két szélső értéke közül az első magasabb.

Írassuk ki a számítógéppel az összes 3-mal osztható számot 666 és 111 között, visszafelé!

```
10 PRINT "□"
20 FOR I=666 TO 111 STEP -3
30 PRINT I,
40 NEXT I
```

Indítsuk el a programot! A számok követhetetlen gyorsasággal rohannak fölfelé, és csak akkor tudjuk egyáltalán megérteni, hogy miről van szó, amikor a futás a végére ér és megáll, akkor viszont már a képernyő tetején eltűnt egy és más.

Hogyan lehet lelassítani a futást? Egyetlen sort kell beleírni a programba, de az annál tanulságosabb lesz.

```
35 FOR N=1 TO 100:NEXT N
```

Hát igen, itt aztán van mit nézni.

Először is: egy programsorba két utasítást is tettünk: egy FORt és egy NEXTet. Eláruljuk: ez teljesen

jogos, a Commodore BASIC két teljes képernyősornyi utasítást elfogad egyetlen programsorként; egy a lényeg: hogy az egy sorba írt különböző utasításokat kettősponttal válasszuk el! (Emlékszünk: ezt a REM utasítás kapcsán már említettük, de most kiderült, hogy nemcsak a REMre igaz, hanem minden BASIC utasításra.)

Másodszor: Nem mondtuk meg a számítógépnek, hogy mit akarunk tőle. Csak annyit mondtunk, hogy 1-től 100-ig csináljon valamit, de hogy mit, azt nem részleteztük.

Nos, épp ez a lényeg. Nem akartunk mást tőle, mint hogy százszor menjen vissza a NEXTtől a FORra. Amíg ezt megcsinálja, addig épp elég idő telik el ahhoz, hogy jól megfigyelhessük, amit kiír. Vagy, ha még mindig túl gyors, akkor a 100-ból csinálhatunk 220-at vagy akár 100000-et is.

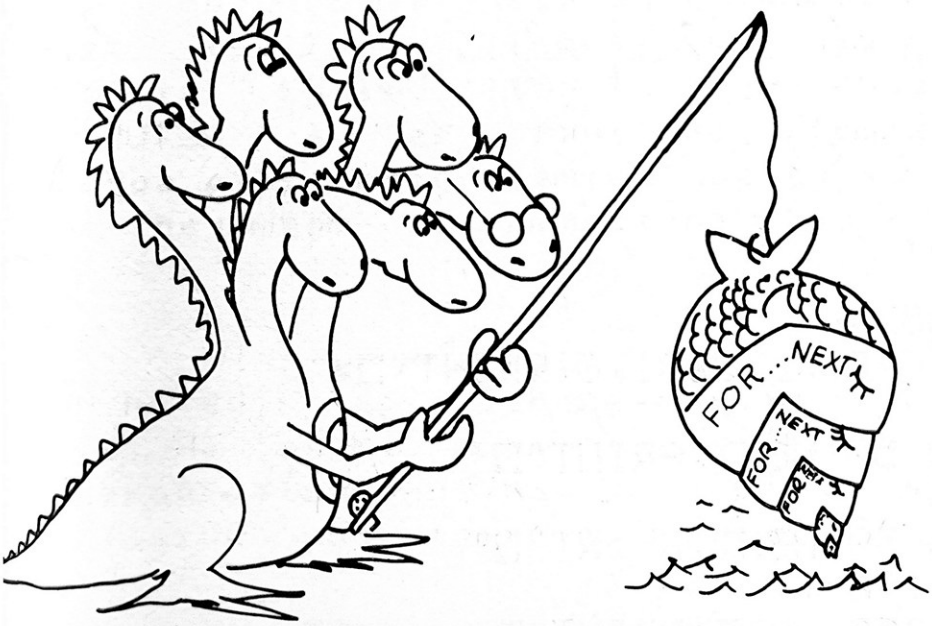
Vizsgáljuk csak meg a programot úgy, ahogy most kinéz!

```
10 PRINT"♥"
20 FOR I=666 TO 111 STEP -3
30 PRINT I;
35 FOR N=1 TO 100:NEXT N
40 NEXT I
```

A 10-es sor ugyebár tisztára söpri a képernyőt. A 20-as megszabja az I értékét, és megmondja, hogy hármasával számoljon a gép visszafelé. A 30-as azt az utasítást adja, hogy a gép írja is ki I értékét, méghozzá egymás mellé. A 35-ös százszorosára növeli azt az időt, amíg a gép eljut a 40-es sorra, ahonnan aztán visszalép a 20-asba.

És itt a következő érdekesség. Ha jól látjuk, itt két FOR... NEXT ciklus van, ráadásul az egyik bele van ágyazva a másikba! A FOR I - NEXT I cikluson belül van a másik, a FOR N - NEXT N.

Jól látjuk, és ez így van rendjén. Két, sőt több FOR... NEXT ciklust is lehet (sőt néha muszáj) használni a BASIC programokban, de lényeges, hogy ezek mindig zártak legyenek, ne kerüljön egyik NEXT se kívül azon a cikluson, amelyiken belül van a hozzá tartozó FOR. Valahogy olyasféleképp kell ezeket szerkeszteni, ahogyan a nagy halak megeszik a kis halakat.




```
FOR A=N TO...
  FOR B=N TO...
    FOR C=N TO...
      FOR D=N TO...
      NEXT D
    NEXT C
  NEXT B
NEXT A
```

És most jöjjön egy feladat: gépeljük be az itt következő programot, és mielőtt lefuttatnánk, próbáljuk meg kitalálni, hogy mit rajzol!

(Figyeljük meg, hogy itt már több, logikailag összetartozó utasítást írtunk egy sorba, így valamicske memóriát takarítunk meg - és persze, ezúttal, papírt is...)

```
10 POKE 53280,6:POKE 53281,6
20 PRINT "♥":PRINT"■":REM CTRL+2
30 PRINT "       \/"
40 PRINT "      ┌──┐"
50 PRINT "      |..|"
60 PRINT "      └──┘"
70 PRINT "        ▌"
80 FOR X=1 TO 5
90 PRINT "     *▒▒▒▒▒*":NEXT X
110 FOR Y=1 TO 4
120 PRINT "     *   *":NEXT Y
```

És még egy ugyanilyen feladat: ideírunk egy programot, és nem mondjuk meg előre, hogy mit csinál. Mindenki próbálja meg kitalálni, aztán gépelje be, futtassa le, és meglátja, hogy helyesen tippelt-e.

```
10 PRINT"♥"
20 FOR I=1 TO 20
30 PRINT I "*" I "=" I*I
40 NEXT I
```

Egy olyan program következik, amelyik egy labdát mozgat a képernyőn balról jobbra. Hosszú lesz, de a begépeléséhez tudunk már egy fogást, amellyel az ismétlődő sorokat csak egyszer kell beírni, utána mindig csak a sorszámot javítjuk, és a RETURN megnyomása után a gép az új sorszámmal hozzáírja a sort a programhoz, de a sor a régi sorszámmal is megmarad a programban.

```
0 REM MOZGO GOLYO
10 POKE53280,6:POKE53281,6:PRINT"□■"
20 PRINT"●"
25 FOR I=1 TO 50:NEXT:PRINT"□"
30 PRINT"   ●"
35 FOR I=1 TO 50:NEXT:PRINT"□"
40 PRINT"      ●"
45 FOR I=1 TO 50:NEXT:PRINT"□"
50 PRINT"         ●"
55 FOR I=1 TO 50:NEXT:PRINT"□"
60 PRINT"            ●"
65 FOR I=1 TO 50:NEXT:PRINT"□"
70 PRINT"               ●"
75 FOR I=1 TO 50:NEXT:PRINT"□"
80 PRINT"                  ●"
85 FOR I=1 TO 50:NEXT:PRINT"□"
90 PRINT"                     ●"
95 FOR I=1 TO 50:NEXT:PRINT"□"
100 PRINT"                        ●"
105 FOR I=1 TO 50:NEXT:PRINT"□"
110 PRINT"                           ●"
115 FOR I=1 TO 50:NEXT:PRINT"□"
120 PRINT"                              ●"
125 FOR I=1 TO 50:NEXT:PRINT"□"
130 PRINT"                                 ●"
```

Ebben a programban tulajdonképpen csak egyszer kell megírni a "fékező" sort, mert mindig ismétlődik, ezért elég a sorszámát növelni. A golyót kirajzoló sor is egyszerűen írható újra meg újra: tízzel megnöveljük a sorszámát, utána a cursorral rászaladunk a golyóra, és háromszor megnyomjuk a SHIFT+INST/DEL billentyűt, majd a RETURNt. Ezt jó néhányszor megismételhetjük, a golyó mindig három hellyel jobbra kerül. Ha finomabbra akarjuk állítani a mozgását, akkor három helyett egy hellyel tegyük odább - viszont akkor a programunk háromszor olyan hosszú lesz.

7. NAP — KIKAPCSOLÓDÁS - DE NEM A GÉPNEK!

A hetedik nap a pihenésé - ez érvényes erre a könyvre is. Ha valaki meg tudja állni, hogy ezen a napon bekapcsolatlanul hagyja a gépet, ám tegye. Bár nemigen hisszük, hogy lesz ilyen...

Az előző fejezetben rajzoltunk egy robotot (ez volt az egyik feladat), most próbáljuk meg megmozgatni, megtornásztatni!

Itt a program hozzá, csak be kell írni:

```
10 POKE 53280,6:REM KEK LESZ A KERET
20 POKE 53281,6:REM KEK LESZ A HATTER
30 PRINT "■":REM FEHER LESZ A ROBOT
40 PRINT "♡":REM URES A KEPERNYO
50 PRINT "         ╲╱":REM SHIFT+M,SHIFT+N
60 PRINT "        ╭──╮":REM SHIFT+U,SHIFT+*,SHIFT+I
70 PRINT "        |..|":REM SHIFT+-
80 PRINT "        ╰──╯":REM SHIFT+J,SHIFT+*,SHIFT+K
90 PRINT "          ▌":REM C=+L,C=+J
100 FOR X=1 TO 5
110 PRINT "       *▒▒▒▒*":REM C=++
120 NEXT X
130 FOR Y=1 TO 4
140 PRINT "        *  *"
150 NEXT Y
160 FOR I = 1 TO 200:NEXT
170 PRINT "♡"
180 PRINT "         ╲╱"
190 PRINT "        ╭──╮"
200 PRINT "        |..|"
210 PRINT "        ╰──╯"
220 PRINT "          ▌"
230 PRINT "*****▒▒▒▒*****"
240 FOR Z=1 TO 4
```

```
250 PRINT "      ▒▒▒▒▒"
260 NEXT Z
270 PRINT "      * *"
280 PRINT "     *   *"
290 PRINT "    *     *"
300 PRINT "   *       *"
310 FOR I=1 TO 200:NEXT
320 GOTO 40
```

Ennek a programnak minden sorát ismerjük már, mindegyikről tudjuk, mit csinál, és miért csinálja.

El lehet játszani vele, hogy lassabban vagy gyorsabban tornázzon a robotunk: ha átírjuk a "fékező" sorokban (amelyekben a FOR és a NEXT között nincs más utasítás) a változó értékének felső határát; meg lehet csinálni, hogy a robot torna közben kacsintgasson (ha a szemet az egyik mozgásfázisban kicseréljük a W billentyűn található, SHIFTtel előhívható grafikus jelre), és még sokféle változtatást tudunk végrehajtani ezen az egyszerű programocskán. A legcélravezetőbb, ha céltudatosan, sorról sorra végigolvassuk, és megpróbálunk a lehető legtöbb helyen változtatni rajta, hogy a végén a tulajdon édesanyja se ismerjen rá!

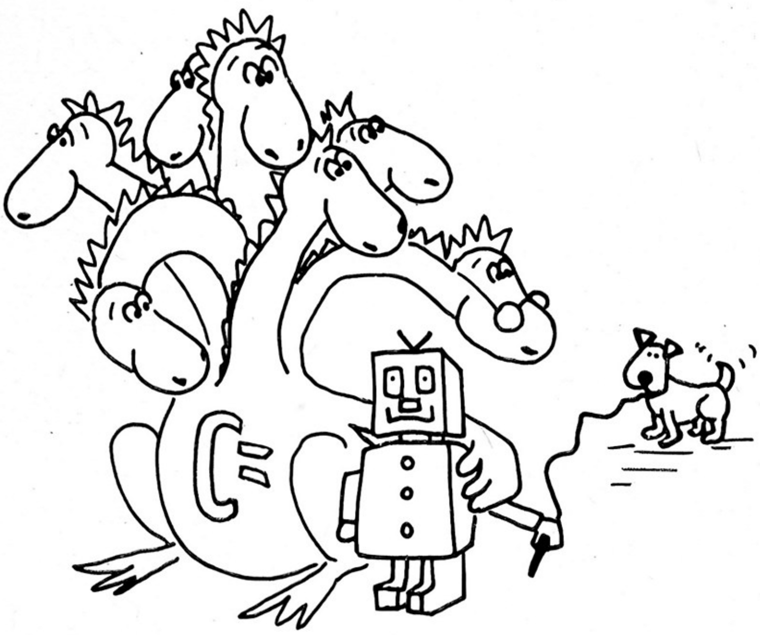
És ha már minden ízében kicseréltük, próbáljuk meg átalakítani mondjuk úgy, hogy egy fülét-farkát billegető kutyát rajzoljon!

A következő program egy valtozekony kedélyállapotú baba: hol jókedvű, hol meg rossz. Próbáljuk meg úgy átírni a programot, hogy többet legyen jókedvű!

```
0 PRINT"♫"
5 POKE 53280,3:POKE 53281,1:PRINT"■♫"
10 PRINT"  ●●●●●●●"
20 PRINT" ●●●●●●●●●"
30 PRINT"●●●     ●●●"
40 PRINT"●●       ●●"
50 PRINT"●●  ▬|▬  ●●"
60 PRINT"●●       ●●"
70 PRINT"●●  O O  ●●"
80 PRINT"●●   ●   ●●"
90 PRINT"●●       ●●"
100 PRINT" ●  ╭─╮  ●"
110 PRINT"  \\   // "
```

```
120 PRINT"    |‿|  "
170 FOR N=1 TO 500:NEXT
500 PRINT"▣":REM CLR/HOME SHIFT NELKUL!
510 PRINT"  ●●●●●●●"
520 PRINT" ●●●●●●●●●"
530 PRINT"●●●     ●●●"
540 PRINT"●●       ●●"
550 PRINT"●● ─  ─  ●●"
560 PRINT"●●  ◯  ◯ ●●"
570 PRINT"●●  ◡  ◡ ●●"
580 PRINT"●●    ●  ●●"
590 PRINT"●● ╱    ╲ ●●"
600 PRINT" ●   ‿   ●"
610 PRINT"  ╲     ╱ "
620 PRINT"    |‿|  "
630 FOR N=1 TO 500:NEXT
670 PRINT"▣"
700 GOTO 10
```

A TÖBBIT A KÖVETKEZŐ KÖTETBEN MONDJUK EL.



TARTALOM

66,– Ft